Contributions à l'arithmétique de la cryptographie

Reynald Lercier

Université d'Aix-Marseille II - Méditerranée
U.F.R. de Sciences

Institut de Mathématiques de Luminy

Contributions à l'arithmétique de la cryptographie

Document de synthèse

présenté et soutenu publiquement le 30 novembre 2004

pour l'obtention de

l'habilitation à diriger des recherches

par

Reynald LERCIER

Composition du jury

Président :	Gilles	LACHAUD
Rapporteurs :	Jean-Marc	COUVEIGNES
	René	SCHOOF
	Serge	VLADUTS
Examinateurs :	Franck	LEPRÉVOST
	Robert	ROLLAND
Directeur de recherche :	Gilles	LACHAUD

Centre d'ÉLectronique de l'ARmement

À mon père, parti trop tôt.

Remerciements

Je remercie en premier lieu Gilles Lachaud pour m'avoir prêté une oreille bienveillante et pour avoir piloté cette habilitation.

Je remercie Jean-Marc Couveignes, René Schoof et Serge Vladuts de me faire l'honneur d'être rapporteurs de ce mémoire. Je suis également grandement honoré de la présence à leurs côtés dans le jury de Gilles Lachaud, Franck Leprévost et Robert Rolland. Merci aussi à Yves Aubry pour avoir su se rendre disponible pour résoudre des questions logistiques ingrates liées à cette habilitation.

Je profite de l'opportunité qui m'est donnée pour rendre hommage à Jean-Marc Couveignes. Ses connaissances scientifiques et ses qualités d'écoute sont pour moi depuis plus de dix ans un enrichissement veritable. J'ai grand plaisir à nos échanges et je lui adresse mes sincères remerciements. Je transmets aussi un grand merci à mon parrain dga, Antoine Joux. J'apprends beaucoup sur la cryptographie à son contact et il m'a accompagné dans nombre de mes travaux.

Je tiens aussi à exprimer ma reconnaissance à mes co-auteurs Antoine Joux, David Lubicz, François Morain, Florent Chabaud, Éliane Jaulmes, Stéphanie Alt et Frédéric Vercauteren, à Stéphane Ballet et Emmanuel Riboulet-Deyris pour les échanges que nous avons eus lors de leur présence au celar ainsi qu'aux personnalités scientifiques de premier ordre que j'ai eu le privilège de côtoyer, Henri Cohen, Bas Edixhoven, Jean-François Mestre et Jacques Stern.

Je suis redevable à la Délégation Générale de l'Armement et en particulier au Centre d'ÉLectronique de l'ARmement de me permettre d'effectuer des travaux de recherche dans de bonnes conditions. Je remercie Paul Pincemin pour avoir compris et encouragé mon choix de carrière. Je remercie également Didier Alquié pour ses multiples qualités qui sont d'une aide quotidienne précieuse et David Lubicz pour sa générosité et son enthousiasme à partager ses connaissances mathématiques.

Je bénéficie quotidiennement de l'influence scientifique stimulante de mes collègues cryptologues, Stéphanie Alt, Anne Babinet, Emmanuel Bresson, Franck Landelle, David Lubicz, Emmanuel Mayer et Valérie Rouat. Plus largement, merci à ceux qui s'investissent pour une activité technique de valeur au celar dans le domaine de la cryptographie. Je pense, entre autres, à Jean-Paul Berlioz, Éric Dupuis, Anthony Lucas, Olivier Mangeot, André Parriel, Pascal Quelen, Michel Vieillard, Philippe Villoing et Brigitte Vital.

Je remercie aussi les partenaires étatiques et industriels du celar avec qui j'ai eu la chance de travailler, en particulier, les laboratoires cryptographiques de la dcssi et de la société thales.

Je remercie chaleureusement ma compagne Murielle pour son soutien. Merci aussi à tous ceux qui m'ont aidé ou qui m'ont encouragé dans mes projets scientifiques.

Avant-propos

Ce document constitue le document soumis en vue de l'obtention du diplôme d'habilitation à diriger des recherches.

La première partie synthétise le principal des travaux de recherches que j'ai menés depuis mon DEA commencé en 1992 [102] jusqu'à maintenant. Ces activités s'inscrivent dans l'intérêt de grande ampleur pour la théorie des nombres et l'arithmétique dans ses aspects calculatoires qui a vu le jour avec l'émergence des systèmes de chiffrement à clef publique. De façon récurrente, les travaux correspondants cherchent à mettre en œuvre efficacement sur ordinateurs des objets mathématiques complexes. Du point de vue cryptographique, de telles constructions peuvent servir aussi bien à définir de nouveaux procédés qu'à montrer les faiblesses d'anciens schémas. Plus précisément, j'organise mon propos au travers cinq chapitres.

- Le premier chapitre est une introduction à la cryptographie à base de courbes elliptiques. Il fait l'objet de la publication [106].
- Le second chapitre explicite les propriétés et les algorithmes nécessaires à la manipulation efficace sur ordinateurs des corps finis. La librairie de calculs arithmétiques `ZEN` [28] qui en résulte est l'un des outils qui fut nécessaire aux expérimentations qui illustrent les chapitres suivants. La description d'un nouveau code d'authentification de message [69] en propose une application originale issue du domaine de la cryptographie symétrique.
- Le troisième chapitre étudie la solidité de l'un des problèmes mathématiques utilisé couramment en cryptographie asymétrique, le calcul de logarithmes discrets dans les corps finis. Les premiers corps étudiés sont ceux de grande caractéristique [79]. Il a été ainsi possible d'effectuer ce type de calcul pour un corps fini dont la caractéristique est un nombre premier de 120 chiffres [76]. Tirant partie de cette expérience, je considère ensuite le cas des corps finis $\mathbb{F}_{2^n}$. Les améliorations obtenues, outre l'aboutissement d'un premier calcul dans $\mathbb{F}_{2^{521}}$ [74], ont fait l'objet de la publication [78].
- Le quatrième chapitre a pour objet l'exemple le plus connu en cryptographie de courbes algébriques, les courbes elliptiques définies sur des corps finis. Le calcul de la cardinalité du groupe défini par ces objets est un préalable incontournable à toute mise en œuvre. Mes actions de recherche en thèse et après thèse concernaient pour l'essentiel ce domaine et sur ce point le plus gros calcul jamais effectué est toujours un calcul dans $\mathbb{F}_{10^{499}+153}$ auquel j'ai pris part en 1995 [112]. Ils ont fait l'objet des publications [111, 110, 103, 105, 113, 114, 75]. Il motive en partie l'un des chapitres d'un livre qui sera publié par CRC Press fin 2004 [32].
- Le cinquième et dernier chapitre s'intéresse à la généralisation aux courbes hyperelliptiques des méthodes connues pour le calcul de la cardinalité de courbes elliptiques. Avec pour point de départ les avancées majeures, initiées par Satoh pour les courbes elliptiques puis par Mestre pour les courbes hyperelliptiques, qui conduisent pour le cas des courbes définies sur les corps finis de petite caractéristique (typiquement $\mathbb{F}_{2^n}$) à des algorithmes de complexité cubique, on montre qu'il est en fait possible de ramener la complexité de ces calculs à une complexité seulement quadratique dans la taille des corps finis [108, 109]. Des calculs pour des courbes de genre 2 dans $\mathbb{F}_{2^{32770}}$ ou de genre 3 dans $\mathbb{F}_{2^{5002}}$, démontrent l'efficacité de ces méthodes [107, 142].

La seconde partie contient la liste de mes publications et les résumés de mes articles classés en quatre thèmes :

- arithmétique des corps finis (page 85),
- le calcul de logarithmes discrets (page 87),
- la cardinalité de courbes elliptiques (page 89) et
- la cardinalité de courbes hyperelliptiques (page 91).

Table des matières

Première partie

Contributions à l'arithmétique de la cryptographie

Introduction

— *Courbes elliptiques et cryptographie*

De nombreux schémas cryptographiques asymétriques peuvent être mis en œuvre dans un groupe fini G dès lors que le problème du logarithme discret y est considéré comme difficile. Rappelons pour un public non-averti que le problème du logarithme discret consiste, étant donné un élément R du sous-groupe de G engendré par un élément P, à trouver l'entier k tel que $R = kP$. L'ensemble des points $E(\mathbb{F}_q)$ d'une courbe elliptique E : $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ définie sur un corps à $q = p^n$ éléments $\mathbb{F}_q$ forme un groupe fini. La loi de groupe est directement dérivée de constructions à la règle qu'il est facile d'illustrer dans le cas réel (cf. Fig. 1).

Mettre en place une cryptographie reposant sur ces courbes est alléchant. En effet, contrairement aux groupes multiplicatifs de $\mathbb{F}_q$, seuls des algorithmes de complexité asymptotique exponentielle en $\log q$ sont connus pour calculer des logarithmes discrets dans $E(\mathbb{F}_q)$. Le plus gros calcul de ce type est un calcul de logarithme discret pour une courbe définie modulo un nombre premier de 109 bits [130]. À niveau de sécurité égal, il semble donc possible d'utiliser des clefs de chiffrement de taille inférieure ($\simeq$ 160 bits) à celles qui sont nécessaires pour $\mathbb{F}_q$ ($\simeq$ 1024 bits).

Ces caractéristiques sont à l'origine d'un engouement récent mais de grande ampleur pour l'utilisation de tels cryptosystèmes en grandeur nature, notamment dans l'élaboration de cartes à puces. Afin de mieux cerner le domaine, nous proposons au lecteur un panorama représentatif que nous avons organisé comme suit : une fois rappelés les premiers articles faisant état de l'utilisation de courbes elliptiques en cryptographie, nous décrivons dans le paragraphe suivant de quelles façons un cryptographe peut construire les courbes dont il a besoin avant de donner ensuite quelques applications.

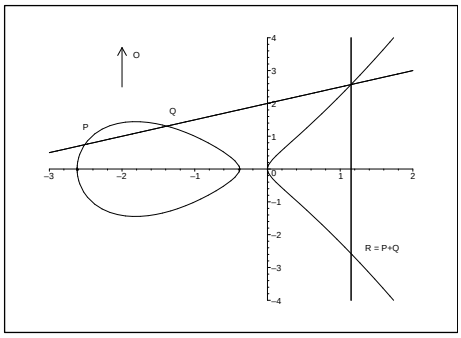


FIGURE 1 – Addition de points P et Q sur la courbe elliptique $E/\mathbb{R} : y^2 = x^3 + 3x^2 + x$.

1 Les fondations de la cryptographie elliptique

Il est généralement admis, qu'indépendamment l'un de l'autre, Miller et Koblitz aient introduit la cryptographie fondée sur les courbes algébriques.

En 1985, Miller [127] part du protocole proposé par Diffie et Hellman [43] pour réaliser un échange de clef en utilisant le groupe multiplicatif défini par un nombre premier p. La sécurité du protocole repose en partie sur l'impossibilité (non-prouvée) de calculer des logarithmes discrets. Les algorithmes alors connus pour cela nécessitaient de l'ordre de $\exp(\sqrt{\log p \log\log p})$ opérations [1]. Miller remarque qu'un tel protocole peut se généraliser à d'autres groupes finis, notamment au groupe des points définis par une courbe elliptique.

Koblitz [87], de son côté, propose à la même époque les analogues des cryptosystèmes à clef publique de Massey-Omura et d'El Gamal pour les courbes elliptiques. Pour la mise en place de ces schémas, il est essentiel de pouvoir trouver des sous-groupes cycliques d'une courbe E dont l'ordre est divisible par un grand nombre premier. Il émet des suggestions variées sur le choix de E et du sous-groupe afin d'obtenir des schémas cryptographiques plus sûrs que ceux obtenus à partir de $\mathbb{F}_q^*$.

2 Quelles courbes elliptiques ?

Pour pouvoir, comme souligné par Koblitz, s'assurer que les courbes elliptiques que l'on utilise en cryptographie ont un ordre divisible par un grand nombre premier il est indispensable de déterminer leur cardinalité. Dans ce domaine, la recherche scientifique a réalisé depuis 1985 d'énormes progrès, d'abord suite aux nombreuses améliorations d'Atkin, Elkies,... qui ont suivies la publication de l'algorithme de Schoof, et plus récemment suite à des idées dues à Satoh et Mestre.

2.1 Courbes obtenues à l'aide du théorème de Weil

Pour une courbe définie sur $\mathbb{F}_q$, il est facile d'obtenir sa cardinalité sur $\mathbb{F}_{q^k}$ ($k \in \mathbb{N}^*$) à partir de celle sur $\mathbb{F}_q$ avec le théorème de Weil [159].

Théorème 1. *Soit E une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ de cardinalité $q+1-c$. Alors la cardinalité de $E(\mathbb{F}_{q^k})$ avec $k \in \mathbb{N}^*$ est égale à $q^k+1-\alpha^k-\beta^k$ où α et β sont les racines complexes du polynôme X^2-cX+q.*

Ainsi, une fois par exemple fixée une courbe E sur $\mathbb{F}_{2^{16}}$ dont il est aisé de déterminer la cardinalité du groupe des points $E(\mathbb{F}_{2^{16}})$ par une recherche exhaustive, il est immédiat de déterminer la cardinalité des groupes $E(\mathbb{F}_{2^{16k}})$.

2.2 Courbes supersingulières

L'inconvénient de la construction précédente est que d'une part, elle restreint le cryptographe à un petit nombre de courbes et que d'autre part, elle n'est d'aucun secours pour des corps finis de grande caractéristique. C'est pourquoi, les chercheurs se sont tout d'abord portés vers une autre famille de courbes, les courbes supersingulières. Ces courbes ont pour particularité que leur cardinalité est égale à p^n+1-c avec $c \bmod p = 0$.

Ainsi, Bender et Castagnoli [12] proposent en 1989 une sous-famille des courbes supersingulières. Comme le calcul de la cardinalité de ces courbes est immédiat, cela permet de répondre facilement aux contraintes cryptographiques, notamment l'utilisation d'une courbe dont l'ordre est le multiple d'un grand nombre premier. Un autre avantage de ces courbes est une légère simplification des opérations de groupe.

Malheureusement, l'usage en cryptographie de telles courbes s'est avéré problématique suites aux travaux menés en 1993 par Menezes, Okamoto et Vanstone [121]. Leur résultat principal montre qu'en toute généralité il est possible de ramener via une réduction appelée "accouplement de Weil" le calcul du logarithme discret dans $E(\mathbb{F}_q)$ à celui du calcul d'un logarithme discret dans $\mathbb{F}_{q^k}$ où $k \in \mathbb{N}^*$. Pour le cas des courbes elliptiques supersingulières, k est égal à 1, 2, 3, 4 ou 6 et la réduction se fait donc en temps probabiliste polynomial en $\log q$, ce qui conduit pour ces courbes à un algorithme probabiliste sous-exponentiel de résolution du logarithme discret.

2.3 Multiplication complexe

Pour pallier l'attaque MOV [121], la communauté s'est intéressé à un autre type de courbes, les courbes "à multiplication complexe" par un ordre dans un corps quadratique $\mathbb{Q}(\sqrt{-D})$ avec D petit. En fait, pour toute courbe elliptique non-supersingulière $E(\mathbb{F}_q)$ de cardinalité $q+1-c$, il existe un entier D sans facteur carré et deux entiers W et V tels que

$$\begin{cases} 4q &= W^2 + DV^2, \\ c &= \pm W. \end{cases}$$

Une des retombées des travaux d'Atkin-Morain dans le domaine de la primalité [7] est une construction de telles courbes pour de "petits" D fixés par avance.

Koblitz [90] explicita en 1991 la construction des courbes à multiplication complexe en caractéristique deux. Les courbes de Koblitz ont les propriétés suivantes :

- elles sont non-supersingulières (l'attaque MOV [121] est donc caduque) ;
- l'ordre du groupe est le multiple d'un grand facteur premier ;
- le doublement de points peut être effectué plus ou moins aussi efficacement que dans le cas des courbes supersingulières ;
- les courbes sont faciles à trouver.

À la même époque, le cas des corps premiers $\mathbb{F}_p$ fut explicité par Morain [132].

Les publications suivantes cherchent à améliorer la mise en œuvre de ces courbes. Miyaji [128], par exemple, cherche à réduire au maximum la taille des données ainsi que les calculs à réaliser sur des cartes à puces pour mettre en œuvre des schémas de signature ou d'authentification à base de courbes elliptiques. Il propose une famille de courbes à multiplication complexe sur $\mathbb{F}_p$ qui respecte ces contraintes tout en conservant un niveau de sécurité suffisamment élevé. Dans le même ordre d'idée, Miyaji souligne dans [129] que l'attaque MOV se complique pour des courbes dont le nombre de points est divisible par la caractéristique p du corps de définition. En fait, des travaux ultérieurs [29] détermineront explicitement le degré de l'extension auquel conduit l'attaque MOV.

2.4 Courbes générales

Si les méthodes précédentes sont efficaces en pratique, il faut tout de même admettre qu'elles restreignent les courbes utilisables. C'est pourquoi les méthodes de calculs *a posteriori* de la cardinalité d'une courbe elliptique fixée *a priori* ont attiré l'attention de la communauté. Cet intérêt était initialement tempéré par la complexité des calculs correspondants mais, après les nombreuses améliorations apportées depuis 15 ans, ces méthodes sont plus que jamais d'actualité.

L'avancée majeure date de 1985. Elle est due à Schoof [152] qui, le premier, exhiba un algorithme dont la complexité est polynomial en $\log q$ pour calculer la cardinalité de $E(\mathbb{F}_q)$, précisément, $O(\log^{5+\varepsilon} q)$. Koblitz [89] décrit dans le détail pour la cas particulier des corps $\mathbb{F}_{2^n}$ l'algorithme de Schoof pour calculer la cardinalité d'une courbe quelconque. Il estimait alors qu'une implantation capable d'effectuer 66000 multiplications dans $\mathbb{F}_{2^{135}}$ en 1 seconde (sur une puce par exemple) serait en mesure de trouver une courbe elliptique dont l'ordre serait 2 fois un nombre premier en environ 5 jours.

Suivent alors des avancées dues à Elkies, Atkin, Couveignes,... qui permettent de faire tomber la complexité à $O(\log^{4+\varepsilon} q)$ [153]. L'algorithme correspondant est appelé l'algorithme SEA. En 1995, on rend compte dans [111] des retombées. Il apparaît notamment que le temps de calcul de la cardinalité d'une courbe définie sur $\mathbb{F}_{2^{155}}$ est réalisable en une dizaine de minutes sur une station de travail classique.

Deux ans plus tard, ce temps tombe à moins d'une minute [105]. De plus, il devient possible d'accélérer la recherche d'une courbe à nombre de points "quasi-premier" en tirant avantage de la structure de l'algorithme de Schoof. Ainsi, on trouve sur $\mathbb{F}_{2^{155}}$ une courbe dont le nombre de point est deux fois un nombre premier en moins d'une heure.

2.5 Corps finis de petite caractéristique $\mathbb{F}_{p^n}$

Au contraire des algorithmes précédents que l'on peut voir comme étant de type "ℓ-adique", une autre voie privilégie pour les corps finis de petite caractéristique des techniques se ramenant à des calculs

dans l'ensemble $\mathbb{Z}_p$ des p-adiques. L'idée commune derrière ces algorithmes est de construire un relevé en caractéristique zéro du Frobenius. Le premier algorithme qui fonctionne plus rapidement que l'algorithme SEA est dû à Satoh [145]. L'idée est de calculer un relevé de E tel que son anneau des endomorphismes est identique à celui de E. Un tel relevé s'appelle le relevé canonique de E. À p fixé (qui doit être petit dans la pratique), le temps de calcul de cet algorithme quand n tend vers l'infini est $O(n^3)$ et la consommation mémoire, $O(n^3)$. Peu de temps après, Vercauteren, Preneel, Vandewalle [168] réduisent l'espace nécessaire à $O(n^2)$. Indépendamment, pour la caractéristique deux, un algorithme de même complexité asymptotique a été découvert par Mestre [124]. Il est basé sur la moyenne arithmético-géométrique (AGM). L'algorithme correspondant (que nous donnons ci-dessous) est remarquablement simple.

Algorithme `AGM`
Cardinalité d'une courbe elliptique $E(\mathbb{F}_{2^n}) : y^2 + xy = x^3 + \alpha$.
INPUT: $\alpha \in \mathbb{F}_{2^n}$.
OUTPUT: L'entier c tel que $2^n + 1 - c$ soit la cardinalité de E.

1. $a := 1 + 8\alpha \in \mathbb{Z}_{2^n}$; $b := 1 \in \mathbb{Z}_{2^n}$;
2. **for** $(i := 1; i < n/2 + 5; i := i + 1)$ {
3. $\quad a, b := \frac{a+b}{2}, \sqrt{ab}$;
4. }
5. A := a ; B := b ;
6. **for** $(i := 1; i < n; i := i + 1)$ {
7. $\quad a, b := \frac{a+b}{2}, \sqrt{ab}$;
8. }
9. **return** $A/a \bmod 2^n \in [-2\sqrt{2^n}, 2\sqrt{2^n}]$.

En 2001, Satoh, Skjernaa et Taguchi [147] améliorent sensiblement la complexité de l'algorithme de Satoh pour atteindre une complexité en temps de $O(n^{2.5})$ et une complexité en espace de $O(n^2)$ après un précalcul en $O(n^3)$.

n	1018	2052	4098	8218	16420	32770	65538	100002
Temps	5 s	20 s	2 mn	10 mn	1 h	6 h	1 j 5 h	3 j 10 h

TABLE 1 – Temps de calcul de la cardinalité de courbes définies sur $\mathbb{F}_{2^n}$ (DEC alpha 613 MHz)

Récemment, on a finalement pu montrer que l'on peut réduire la complexité du calcul du nombre de points à $O(n^2)$ [108, 62]. Pour comparer avec les méthodes ℓ-adiques, le temps de calcul de la cardinalité d'une courbe sur $\mathbb{F}_{2^{155}}$ est maintenant inférieur à la seconde. À titre indicatif, on donne des temps de calcul sur station DEC Alpha pour des corps finis de grandes tailles (cf. Tab. 1).

Remarquons cependant que les courbes définies sur $\mathbb{F}_{p^n}$ peuvent être sensibles à l'attaque dite de "descente de Weil" qui permet de ramener le logarithme discret sur courbes elliptiques définies sur une extension $\mathbb{F}_{p^n}$ d'un corps fini petit $\mathbb{F}_{p^m}$ à celui du logarithme discret pour une courbe de plus grand genre, mais définie seulement sur $\mathbb{F}_{p^m}$. Elle fut principalement proposée par Frey en 1998, approfondie par Galbraith et Smart en 1999, puis rendue efficace par Gaudry, Hess et Smart en 2000. Cette dernière est néanmoins inopérante lorsque n est un nombre premier.

3 Quelle cryptographie ?

Bien sûr, tout système cryptographique reposant sur le problème du logarithme discret s'étend au cas des courbes elliptiques. Nous nous concentrons ici aux résultats spécifiques pour trois types de cryptosystèmes : les générateurs pseudo-aléatoires, les analogues de Diffie-Hellman et les analogues de RSA. Mais plus originalement, la difficulté à inverser des accouplements bilinéaires définis à partir de courbes elliptiques à conduit à de nouvelles applications cryptographiques. Elles font brièvement l'objet d'un dernier paragraphe.

3.1 Générateurs pseudo-aléatoires prouvés

Kaliski présente dans [81] un générateur pseudo-aléatoire qui produit des bits qui sont aussi difficiles à deviner que de résoudre le problème de logarithme discret sur une courbe elliptique. Notons que les courbes qu'il préconise sont des courbes supersingulières.

Quelques permutations définies à l'aide de courbes elliptiques sont aussi présentées dans [82]. Kaliski espère que la majeure partie de ces permutations sont des fonctions à sens unique. Les courbes elliptiques $y^2 \equiv x^3 + b \bmod p$, avec $p \equiv 2 \bmod 3$, $y^2 \equiv x^3 + ax \bmod p$, avec $p \equiv 3 \bmod 4$ et $(a/p) = 1$ ou les paires composées d'une courbe elliptique et de sa tordue, $y^2 \equiv x^3 + ax + b \bmod p$, $y^2 \equiv x^3 + au^2x + bu^3 \bmod p$, avec $(u/p) = -1$ sont utilisées dans la construction.

Pour le petit nombre de courbes elliptiques E définies sur $\mathbb{F}_2$, mais regardées comme des courbes définies sur une extension $\mathbb{F}_{2^n}$, Meyer et Staffelbach développent un nouvel algorithme pour calculer des multiples d'un point arbitraire de E [119]. L'algorithme est présenté comme trois fois plus rapide que la méthode binaire usuelle, plus facile à implanter et ne nécessite aucun précalcul ou mémoire additionnelle. Il est utilisé pour calculer des permutations à sens unique efficaces mettant en jeux des paires composées d'une courbe elliptique et sa tordue en généralisant la construction de [82] aux corps de caractéristique deux.

3.2 Les analogues de RSA

Le préalable à une cryptographie de type RSA est d'exhiber des fonctions à sens unique à trappe. C'est ce que font en 1991 les auteurs de [93]. En fait, trois nouvelles familles de fonctions, basées sur des courbes elliptiques définies sur $\mathbb{Z}/N\mathbb{Z}$ où N est composé, sont proposées. La première famille est une construction naïve uniquement utilisable dans un schéma de signature. La seconde famille peut se substituer aux fonctions à sens unique de type RSA. La troisième famille a les mêmes caractéristiques que les fonctions à sens unique de Rabin. La sûreté de ces fonctions repose sur la difficulté de factoriser N.

Demytko [41] étend les résultats précédents. Il amoindrit les restrictions relatives aux types de courbes elliptiques et de nombres premiers utilisables. D'autre part, il évite les problèmes liés au plongement de messages dans la courbe en travaillant sur une structure de groupe multiple, ce qui permet des opérations de cryptage et de décryptage uniquement basées sur l'abscisse des points de la courbe.

En 1995, Koyama [92] propose un système cryptographique à clef publique reposant, non pas sur des courbes elliptiques, mais sur les courbes singulières $y^2 + axy = x^3$ définies sur $\mathbb{Z}/N\mathbb{Z}$. Les caractéristiques de ce schéma sont similaires à celle de RSA, excepté, peut-être, une rapidité accrue. Peu après, les auteurs de [126] utilisent les mêmes idées que Williams [172] pour obtenir un système dont il est, cette fois, prouvé que toute attaque est *équivalente* à la factorisation d'entiers.

Une analyse plus fine de ses idées impose néanmoins une certaine prudence. Par exemple, Hastad [66] a montré qu'utiliser de faibles exposants e dans RSA n'est pas sûr si un même message est chiffré à destination de multiples destinataires. Cela est vrai même si une estampille datée est utilisée pour chaque receveur. Si par exemple $e = 3$ et le nombre de destinataires est 7, l'attaquant peut retrouver le message en clair à partir des 7 cryptés. De même, les auteurs de [94] montrent que les cryptosystèmes à base de courbes elliptiques précédents ne sont pas sûrs si $e = 5$, si N est supérieur à 2^{1024} et le nombre de destinataires est égal à 428. Dans la version de Demytko, e peut prendre la valeur 2. Dans ce cas, le schéma n'est pas sûr si le nombre de receveurs est 11 pour $N \geq 2^{175}$.

En fait, utiliser un analogue du RSA sur courbes elliptiques plutôt que le RSA lui-même ne semble pas clairement plus avantageux. Nous renvoyons le lecteur aux travaux de Joye [80] pour un panorama complet sur la question.

3.3 Les analogues de Diffie-Hellman

La majeure partie de la cryptographie courbes elliptiques d'aujourd'hui repose sur les idées de Diffie-Hellman et utilise le fait que le problème du logarithme discret est réputé difficile sur ces groupes. En retracer l'historique dépasse largement la portée de ce texte. Disons simplement que l'une des avancées majeures de la cryptographie moderne est la recherche de preuves de sécurité pour les protocoles

cryptographiques. Une idée récurrente consiste à supposer qu'un attaquant est capable de casser le cryptosystème étudié et d'en déduire un algorithme capable de résoudre le problème mathématique sous-jacent. S'il se trouve par ailleurs que ce problème est réellement difficile, alors on en déduit par la contraposée qu'un tel attaquant ne peut exister. La spécificité liée au fait que l'on utilise dans un protocole une courbe elliptique est généralement gommée en modélisant ce groupe comme un "groupe générique". Nous renvoyons le lecteur intéressé par ces sujets à l'excellente synthèse réalisée dans le cadre du projet européen NESSIE [141].

3.3.1 Signature.

L'algorithme de signature électronique le plus connu est l'algorithme ECDSA. Sa sécurité repose sur l'impossibilité de résoudre le logarithme discret dans un sous-groupe d'une courbe elliptique. Il a été proposé en premier par Vanstone en 1992 en réponse à une requête du NIST [165]. C'est un standard ISO depuis 1998, un standard ANSI depuis 1999 et un standard IEEE et NIST depuis 2000.

Excepté une tentative récente de preuve dans le modèle du groupe générique [20], il n'existe pas de preuve de sécurité pour cet algorithme. Seules des variantes semblent pour le moment en bénéficier.

3.3.2 Chiffrement asymétrique.

Une façon usuelle d'utiliser un mécanisme de chiffrement asymétrique est de se servir du mécanisme pour chiffrer uniquement une clef (*i. e.* un message de petite taille) et ensuite se servir de celle-ci pour chiffrer par un algorithme de chiffrement symétrique le message proprement dit.

Une formalisation récente de ces protocoles due à Cramer et Shoup [40] s'appelle le modèle "KEM-DEM" ("Key/Data Encapsulation Mechanism"). Cette formalisation permet plus facilement d'exhiber des preuves de sécurité sur les protocoles.

3.4 Les systèmes à clefs publiques arbitraires

Dans une infrastructure asymétrique classique, les clefs publiques sont issues des clefs privées. Il n'existe en fait que depuis peu une solution réellement satisfaisante pour construire des systèmes dans lesquels tout identifiant puisse être la clef publique d'une clef privée calculable *a posteriori*.

Une première tentative reposait sur une variation de l'échange de clef Diffie-Hellman due à Mc Curley. Dans ce protocole, un attaquant doit à la fois casser le problème de Diffie-Hellman tout en factorisant de grands entiers. Il est fait état dans [162] d'une généralisation de ce schéma aux courbes elliptiques.

Très récemment, on a pris conscience que l'accouplement de Weil, initialement utilisé dans l'attaque MOV [121], peut à l'inverse être mis à profit pour la construction. Les fondements de cette nouvelle approche est un article de Joux publié en 2000 [70] où il est expliqué qu'il est possible à trois partenaires de faire un échange Diffie-Hellman en une passe à l'aide de l'accouplement de Weil. Avec cette brique de base, Boneh et Franklin [18] ont proposé un schéma de chiffrement asymétrique pour lequel n'importe quelle chaîne de caractères peut être une clef publique. Le schéma correspondant est prouvé sûr dans le modèle de l'oracle aléatoire et en supposant une variante calculatoire du problème Diffie-Hellman pour courbes elliptiques. Une messagerie construite sur ces idées aurait des propriétés fonctionnelles intéressantes. Les abonnés pourraient en particulier envoyer du courrier à des destinataires qui ne disposent pas encore de clef privée.

Depuis, cette cryptographie a littéralement explosée. La plupart des fonctionnalités habituellement rendues par la cryptographie asymétrique peuvent l'être par des schémas qui reposent sur l'accouplement de Weil ou de Tate. Un panorama de ce domaine en plein mouvement peut par exemple être trouvé à [8].

4 Les applications

Les premières expérimentations pratiques mettant en œuvre de la cryptographie à base de courbes elliptiques sont des résultats de laboratoire.

Ainsi, en 1989, Mullin et Vanstone [2] présentent pour la première fois une carte électronique implantant des opérations sur des courbes elliptiques [2]. Cette carte était construite autour d'un processeur

Motorola M68008. En 1992, la faisabilité de cryptosystèmes construits à base de courbes elliptiques avec des clefs de l'ordre de 100 bits est étudiée dans [64]. Des taux de 2Kbits/s sont atteints sur station de travail. En 1993, Menezes et Vanstone [122] explorent la faisabilité d'implanter en hardware un processeur arithmétique pour effectuer des calculs sur des courbes elliptiques définies sur des corps finis de caractéristique deux. Enfin, en 1995, Schroeppel et al. [155] décrivent une implantation logicielle d'un échange de clef à la Diffie-Hellman à base de courbes elliptiques définies sur $\mathbb{F}_{2^{155}}$ dont les performances sont, à niveau de sécurité équivalent, légèrement meilleures que celles du Diffie-Hellman usuel.

Depuis ces expérimentations, plusieurs normes déjà approuvées ou en cours d'approbation ont vues le jour. Nous rappelons les principales ci-dessous.

- ANSI, "American National Standards Institute".

- `ANSI X9.62` : "Elliptic Curve Digital Signature Algorithm " (ECDSA) est un standard de signature électronique.
- `ANSI X9.63` : "Elliptic Curve Key Agreement and Key Management" normalise l'utilisation de courbes elliptiques à des fins de chiffrement.

- FIPS, "Federal Information Processing Standard" du NIST (US government's "National Institute of Standards and Technology"). Il s'agit d'une extension du standard de signature électronique (DSS) pour inclure l'algorithme ECDSA (`ANSI X9.62`).

- IEEE. La norme IEEE 1363-2000 couvre la cryptographie asymétrique à base de logarithmes discrets (corps finis ou courbes elliptiques) ainsi que celle à base de RSA. Elle spécifie les mécanismes ECDSA, ECDH et ECMQV.

- PKCS, émis par la société RSA inc. Cette norme adresse de nombreux aspects : la génération des clés et des paramètres, la signature électronique, le chiffrement asymétrique, etc.

5 Conclusion

D'abord l'apanage des mathématiciens avant de devenir un sujet d'étude privilégié pour les cryptographes, les courbes elliptiques commencent à être connues d'un plus large public. Gageons que dans un monde dominé par la cryptographie à clef publique RSA, ces dernières deviennent à terme une alternative crédible.

Arithmétique des corps finis

— *Applications à la cryptographie symétrique*

Au carrefour de l'arithmétique où la notion de corps est centrale et de l'informatique où la notion de finitude est inhérente à tout algorithme, les corps finis sont au cœur des travaux de cette synthèse. Ils ont motivé avec le concours de Chabaud, le développement d'une librairie de calcul, la librairie `ZEN` [28] dont le but premier est la manipulation aisée et efficace des corps finis sur ordinateur.

Si du point de vue mathématique, la théorie des corps finis est finalement assez simple, il est assez déconcertant de constater que leur mise en œuvre efficace sur ordinateurs fait appel à une multitude de techniques qui, aujourd'hui encore, continuent à s'enrichir. Nous développons cet aspect en première partie de ce chapitre (cf. paragraphe 1). Dans une seconde partie, nous proposons, issu des travaux du domaine de la cryptographie symétrique menés avec Jaulmes [69], un nouveau code d'authentification de message dont les performances dépendent directement de l'arithmétique dans les corps finis utilisée, le schéma FRMAC (cf. paragraphe 2).

1 Corps finis

On introduit les quelques notions d'algèbre élémentaire sur les corps commutatifs nécessaires à la compréhension des algorithmes du paragraphe 1.3. De nombreux textes existent sur le sujet, citons parmi d'autres [115, 133].

1.1 Les corps commutatifs en général

Un anneau dans lequel tout élément non-nul admet un inverse à gauche pour la multiplication est un corps. Un corps est dit commutatif si sa loi de multiplication est commutative. Sauf mention contraire, c'est toujours le cas ici. Soit L un corps contenant un sous-ensemble K qui lui-même est un corps pour les opérations héritées de L, alors L est appelé une extension de K, et K, un sous-corps de L. Chaque corps a un plus petit sous-corps, appelé le sous-corps premier, qui est isomorphe soit au corps des rationnels $\mathbb{Q}$ (on parle de corps de caractéristique 0), soit à un corps à p éléments avec p un nombre premier (on parle de corps de caractéristique p).

Si L est une extension de K, et θ est un élément de L hors de K, alors le plus petit sous-corps de L contenant K et θ est noté $K(\theta)$. De même, le plus petit corps contenant K et les éléments θ_1, θ_2, …, θ_n de L est noté $K(\theta_1, \theta_2, \ldots, \theta_n)$. Toute extension L de K peut être regardée comme un espace vectoriel sur K. Sa dimension est appelée le degré de L sur K, elle est notée $[L : K]$. Si cet espace vectoriel est de dimension finie, L est dite une extension finie de K. Au passage, si L est une extension finie de K et M est une extension finie de L, alors $[M : K] = [M : L] \times [L : K]$.

Soit $K[x]$ l'anneau des polynômes à indéterminée x et à coefficients dans K, alors $K[x]$ est un anneau principal. Si $f(x)$ est un polynôme irréductible de $K[x]$, c'est-à-dire s'il n'est pas le produit de deux polynômes non-constants de $K[x]$, alors il existe une extension L de K de degré minimal qui contient ses racines. Enfin, si θ est l'une des racines de $f(X)$, $\{1, \theta, \theta^2, \ldots, \theta^{[L:K]-1}\}$ forme une base de $K(\theta)$ sur K. Notons aussi que deux corps de décomposition d'un même polynôme sont isomorphes.

L'ensemble des automorphismes d'un corps forme un groupe pour la composition. Si σ_1, σ_2, …, σ_n sont des isomorphismes distincts de K_1 dans K_2, alors ces isomorphismes sont linéairement indépendants

sur K_2 dans le sens où, si $\sum a_i \sigma_i(b) = 0$ pour tout b dans K_1 avec les a_i dans K_2, alors les a_i doivent être nuls (lemme de Dedekind). Un automorphisme de corps laisse les éléments du sous-corps premier invariants. Plus généralement, l'ensemble des éléments invariants par un sous-ensemble à n éléments du groupe des automorphismes de K est un sous-corps de K et on peut montrer que la dimension de K par rapport à ce corps est supérieure ou égale à n.

1.2 Corps finis

Les corps finis sont, comme leur nom l'indique, des corps avec avec un nombre fini d'éléments. Comme ils ne peuvent pas contenir $\mathbb{Q}$, ils sont toujours de caractéristique p ou p est un nombre premier. Ils ont p^n éléments où n est un entier strictement positif. Comme à l'inverse le corps de décomposition du polynôme défini sur $\mathbb{F}_p$ par $x^{p^n} - x$ a p^n éléments, il y a pour tout p^n, à isomorphisme près, un unique corps fini à p^n éléments.

Une importante propriété des corps finis est que leur groupe multiplicatif est aussi cyclique. Un générateur est appelé un élément primitif. Une conséquence est que les sous-corps de $\mathbb{F}_{p^n}$ sont les corps $\mathbb{F}_{p^m}$ pour les diviseurs m de n.

Enfin, le groupe des automorphismes d'un corps fini est aussi cyclique. Un générateur de ce groupe est l'automorphisme Frobenius défini pour un corps fini de caractéristique p par $x \longrightarrow x^p$.

1.3 Multiplication rapide dans les corps finis

La librairie `ZEN` vise à la mise en œuvre efficace d'opérations arithmétiques pour corps finis sur ordinateurs généralistes. Pour cette cible, la représentation naturelle d'un corps fini $\mathbb{F}_{p^n}$ via une base polynomiale $\{1, \theta, \theta^2, \ldots, \theta^{n-1}\}$ s'avère la plus efficace. Si dans de telles bases, la loi d'addition ne pose pas de problème car les algorithmes classiques sont déjà de complexité linéaire en la taille du corps, une difficulté majeure concerne par contre la loi de multiplication car la mise en œuvre naïve conduit à des temps quadratiques en la taille des objets.

Un modèle d'évaluation de la complexité de ces algorithmes assez représentatif de ce que l'on observe en pratique est celui d'une machine de Turing à plusieurs bandes [163], codant une opération `MUL` calculant sur $2B$ bits le produit de deux entiers simple précision de B bits où B vaut typiquement 32 ou 64. Un algorithme sera considéré d'autant plus efficace que le nombre d'opérations effectuées par la machine de Turing, sa complexité en temps, sera bas. À algorithmes similaires pour ce critère, un algorithme sera considéré plus efficace que l'autre si le nombre d'accès mémoire qu'il effectue est moindre. Avec cette mesure, nous donnons ci-dessous des algorithmes qui sont, selon les corps finis considérés, parmi les plus efficaces pour réaliser la multiplication de deux éléments.

Une première référence sur le sujet est bien sûr le texte de Knuth [86]. Une synthèse plus récente est due à Bernstein [15]. Aussi, une mise en œuvre sérieuse de certains de ces algorithmes, dont un compte rendu peut-être trouvé dans [63], a été réalisée pour la librairie de calcul GMP [57].

Remarque 1. *Des modèles de machines plus puissantes que les machines de Turing existent, aussi bien d'un point de vue théorique que pratique ("Randoms Access Machines", circuits intégrés "VLSI", etc.). Des mesures de complexité alternatives existent pour ces modèles (complexité bilinéaire [31], temps par espace [14], etc.) qui conduisent à des algorithmes autres plus adaptés mais, en dehors des travaux menés par l'auteur, nous n'y faisons pas mention ici.*

1.3.1 Corps finis de petite taille

- Les corps $\mathbb{F}_p$, $p < 2^B$. Chaque élément est représenté comme un entier simple précision (d'au plus B bits). La multiplication modulaire est une simple opération `MUL` suivi d'une division euclidienne du résultat par p. Cette division étant souvent longue sur les ordinateurs actuels, il est souvent plus astucieux de la remplacer par deux appels à `MUL` une fois précalculé le quotient par p de la puissance de deux immédiatement supérieure à p^2 [9].

- Les corps $\mathbb{F}_{2^n}$, $n < B$. La méthode la plus rapide consiste à construire à partir des bits des éléments de $\mathbb{F}_{2^n}$ dont on veut le produit, des entiers dont un bit sur deux est nul et de multiplier ces entiers avec `MUL`. La réduction peut être réalisée par une division euclidienne où tout comme $\mathbb{F}_p$, réalisée par deux opérations `MUL` une fois le polynôme réciproque du polynôme de définition de $\mathbb{F}_{2^n}$ précalculé.

- Les corps $\mathbb{F}_{p^n}$, p impair. Les ordinateurs étant mal adaptés à des objets non-binaires ou non-entiers, le plus simple est de finalement tabuler les opérations.

Pour $p^n \leqslant 2^8$, on associe aux éléments de ces corps des indices compris entre 0 et $p^n - 1$ et on précalcule la table de multiplication associée.

Pour $p^n \leqslant 2^{16}$, les corps finis étant multiplicativement cycliques, nous représentons chaque élément y par son indice x si $y = g^x$ (g une racine primitive de $\mathbb{F}_{p^n}$). Le produit de deux éléments est alors la somme des indices modulo $p^n - 1$ (méthode de Zech, cf. [115]).

1.3.2 Corps finis de taille moyenne

- Les corps $\mathbb{F}_p$. La méthode habituelle de représentation des entiers positifs est de les écrire en base 2^B. Un entier est alors stocké sur ordinateur comme un vecteur d'entiers en simple-précision, les composantes de la représentation en base 2^B.

La méthode usuelle, celle que nous avons tous appris à l'école primaire est dans les faits la plus rapide pour des tailles moyennes. Elle consiste à faire tous les produits simple précision `MUL` des composantes en base 2^B des entiers à multiplier et à additionner les résultats à la bonne position dans le produit final. Sa complexité est de $O(\log^2 p)$. La réduction modulaire peut-être réalisée de façon similaire. Itérativement, on approxime le quotient des $2B$ bits de poids fort du produit par les B bits de poids fort du diviseur et on soustrait à un accumulateur le produit du diviseur avec le quotient. Sa complexité est de $O(\log^2 p)$.

Le quotient simple précision étant souvent une opération bien plus longue que `MUL` sur les microprocesseurs modernes, la division euclidienne prend généralement beaucoup plus de temps que le produit, c'est pourquoi des méthodes alternatives ont été proposées. La méthode qui vient immédiatement à l'esprit consiste à précalculer le quotient par p d'une puissance de deux plus grande que p^2 et de remplacer la division euclidienne par des produits. Une mise en œuvre astucieuse de ce procédé est due à Barrett [9].

Une autre méthode très populaire est la réduction dite de de Montgomery [131]. Soient R une puissance de 2 supérieure à p (supposé impair) et T un entier tel que $0 \leq T < pR$, alors Montgomery a proposé une méthode qui permet de calculer efficacement $TR^{-1} \bmod p$. Supposons ainsi que l'on dispose de la représentation de Montgomery $\tilde{x}$ et $\tilde{y}$ de deux entiers x et y, i. e. $\tilde{x} = xR \bmod p$ et $\tilde{y} = yR \bmod p$, alors la représentation de Montgomery du produit xy est égale à la réduction de Montgomery du produit $\tilde{x}\tilde{y}$, c'est-à-dire $(\tilde{x}\tilde{y})R^{-1} \bmod p$.

- Les corps $\mathbb{F}_{2^n}$. Avec pour brique élémentaire le produit de deux polynômes de degrés bornés pas B (cf. paragraphe 1.3.1), adapter les méthodes que nous venons de décrire pour $\mathbb{F}_p$ au cas des corps de caractéristique 2 n'est pas difficile, et ce d'autant plus que les additions modulo 2 se vectorisent aisément sur ordinateurs.

- Les corps $\mathbb{F}_{p^n}$, p impair. La méthode immédiate est d'utiliser une arithmétique polynomiale construite au dessus d'opérations arithmétiques pour $\mathbb{F}_p$.

Lorsque p est petit, cela est inefficace. Lorsque n est composé, une stratégie possible est de remplacer l'arithmétique modulo p par une arithmétique pour $\mathbb{F}_{p^m}$ avec m un petit facteur de n. Sinon, une stratégie alternative est de construire à partir des éléments $\sum_i a_i\theta^i$ écrits dans une base polynomiale θ dont on souhaite le produit, des entiers dans lesquels on intercale entre les bits correspondants aux a_i autant de bits nuls, d'effectuer un produit d'entiers et d'en extraire le polynôme résultat.

1.3.3 Corps finis de taille grande à gigantesque

- Les corps $\mathbb{F}_p$. D'abord considérées d'un point de vue plus théorique que pratique, les méthodes asymptotiquement les plus rapides pour réaliser le produit de deux entiers sont devenues depuis une demi-douzaine d'année incontournables pour les calculs dans des corps finis dont la taille est de plusieurs milliers à plusieurs milliards de bits. Une idée récurrente derrière ces méthodes est de remplacer 2^B par une indéterminée X dans l'écriture en base 2^B des entiers et de ramener le problème du produit d'entiers à celui de polynômes. Le produit peut alors être réalisé, d'abord par l'évaluation des opérandes en des points bien choisis à partir desquels il est facile d'obtenir l'évaluation du polynôme produit, puis par l'interpolation du produit des évaluations précédentes pour finalement obtenir le produit recherché.

L'algorithme de Karatsuba. Cet algorithme [83] est la première méthode asymptotiquement plus rapide que la méthode de multiplication classique. Étant donné deux polynômes $a_0 + a_1X$ et $b_0 + b_1X$, le produit $c_0 + c_1X + c_2X^2$ peut-être obtenu par $c_0 = a_0b_0$, $c_2 = a_1b_1$ et $c_1 = (a_0 + b_0)(a_1 + b_1) - c_0 - c_2$. Appliquer cette idée à la multiplication d'entiers, le produit de deux entiers de taille $\log p$ se ramène au produit de trois entiers de taille $\frac{1}{2}\log p$. Si on réitère récursivement la méthode, on obtient finalement une complexité de l'ordre de $O(\log^{\log_2 3} p)$.

En pratique [57], la méthode de Karatsuba est plus rapide que la méthode classique pour des entiers de l'ordre de 2000 bits (à moduler bien sûr selon les microprocesseurs cibles).

L'algorithme de Toom. Reconsidérant la méthode de Karatsuba sous l'angle évaluation/interpolation, Toom [164] généralise la méthode Karatsuba en découpant les opérandes en plus que deux morceaux. Améliorée par Cook [33] puis par Knuth [86], la complexité obtenue, en considérant un nombre de morceaux croissant asymptotiquement avec $\log p$, est $O((\log p)^{1+\sqrt{2/\log\log p}} \log\log p)$.

En pratique [57], la méthode de Toom est plus rapide que la méthode de Karatsuba pour des entiers de l'ordre de 5000 bits (à moduler aussi selon les microprocesseurs cibles).

Transformée de Fourrier rapide. L'utilisation de points d'interpolations astucieux conduit à la transformée de Fourier rapide. Schönhage et Strassen [151] sont ainsi les premiers à proposer une méthode dont la complexité est bornée par $(C\log_2 p)(C\log_2\log_2 p)(C\log_2\log_2\log_2 p)\cdots$ où C est une constante et où le dernier terme est tel que $\log_2 \ldots \log_2 p \leq 2$. Ils amélioreront ensuite l'algorithme en utilisant la transformée de Fourrier modulo des entiers de Fermat $(2^{2^k}+1)$ [3]. L'algorithme correspondant est l'algorithme de multiplication asymptotiquement le plus rapide connu à ce jour. Il a complexité $O(\log p \log\log p \log\log\log p)$.

En pratique [57], la méthode de multiplication par FFT est plus rapide que la méthode de Toom pour des entiers de l'ordre de 150000 bits (à moduler toujours selon les microprocesseurs cibles).

- Les corps $\mathbb{F}_{p^n}$ avec $n > 1$. La stratégie la plus immédiate pour retrouver dans le cas des extensions de corps des complexités du même ordre que pour $\mathbb{F}_p$ est comme précédemment de construire des entiers dont certains des bits sont les bits des polynômes à multiplier, d'utiliser l'une des méthodes de multiplication d'entiers par FFT et de retrouver ensuite dans le produit le polynôme résultat.

Pour des corps de caractéristique p petite, notons cependant qu'il est possible d'adapter directement les algorithmes de multiplications précédents [150]. Citons par exemple des travaux de Cantor [22] qui conduisent à caractéristique p fixe à une complexité asymptotique égale à $O(n(\log n)^{1+\log_p((p+1)/2)})$.

2 Une application : FRMAC

Quand des clefs symétriques partagées sont disponibles, l'utilisation de MACs (codes d'authentification de message) est probablement la façon la meilleure pour protéger des informations en intégrité. Dans ce domaine, l'approche de Wegman-Carter est une démarche qui conduit à des constructions efficaces dont la sécurité peut être prouvée [24, 25]. Elle consiste à d'abord utiliser une fonction de hachage efficace paramétrée par une clef, h_k, membre d'une famille de hachage dite quasi ε-universelle, afin de réduire le

message $\mathcal{M}$ à une taille fixe, puis à l'addition modulo deux du résultat du hachage et l'image d'un aléa une fois (un "nonce") R par une fonction pseudo-aléatoire paramétrée aussi par une clef, f_K. Le MAC résultant peut ainsi être résumé comme suit, $f_K(R) \oplus h_k(\mathcal{M})$.

Le plus efficace de ces MACs actuellement connus pour une cible logicielle pour de grands messages est le schéma UMAC du à Black, Halevi, Krawczyk, Krovetz, et Rogaway (cf. [138] pour une comparaison sur le comportement de plus vieilles constructions en logiciel). Ce MAC, d'abord proposé en 1999 [16], a évolué en 2000 vers une construction plus mature appelé UMAC-2000 qui a été soumise par ses auteurs au projet européen NESSIE [141]. Les familles de hachage de UMAC-2000, UHash 16 ou UHash 32, sont incroyablement efficaces sur ordinateurs modernes, facilement parallélisables et leurs probabilités de collision peuvent être adaptées facilement aux besoins utilisateur.

Une caractéristique agréable des MACs Wegman-Carter est que leurs preuves de sécurité sont facilement compréhensibles. Elles indiquent principalement que la probabilité de contrefaçon d'un agresseur ne peut pas dépasser la somme de deux termes. Le premier terme correspond à l'avantage pour distinguer une famille F de fonctions à sortie n bits de fonctions pseudo-aléatoires parfaites. Le seconde terme est le nombre de tentatives de contrefaçons, noté q, multiplié par la probabilité de collision ε. Ceci produit une borne de la forme $\mathbf{Adv}_F^{\mathbf{prf}}(\mathcal{A}) + q(\varepsilon + 1/2^n)$ [16]. Le premier terme dépend de F. Pour une famille de permutations cryptographiques de $\{0,1\}^n$, par exemple l'algorithme de chiffrement par bloc AES [139], on peut supposer classiquement que $\mathbf{Adv}_F^{\mathbf{prf}}(\mathcal{A}) \simeq q^2/2^{n+1}$ [10]. Le deuxième terme est proche de l'avantage d'un agresseur dont la stratégie consiste à vérifier une étiquette engendrée au hasard correspond à celle d'un message $\mathcal{M}$.

En dépit de la simplicité indéniable de la théorie sous-jacente, quelques difficultés peuvent cependant apparaître quand on utilise un tel MAC en pratique. Nous en énumérons deux ci-dessous.

- Il est crucial dans la preuve de sécurité que tous les nonces soient distincts. Si, par exemple, un nonce R est utilisé pour une étiquette $T_{\mathcal{M}}$ d'un message $\mathcal{M}$ et si pour l'étiquette $T_{\mathcal{N}}$ d'un autre message $\mathcal{N}$, alors une étiquette $T'_{\mathcal{M}}$ pour $\mathcal{M}$ avec le nonce $R' \neq R$ peut être forgée en une étiquette pour $\mathcal{N}$ avec nonce R', $T'_{\mathcal{N}} = T'_{\mathcal{M}} \oplus T_{\mathcal{M}} \oplus T_{\mathcal{N}}$. Dans une application où plusieurs utilisateurs partagent une même clef, ce peut être une vraie difficulté pour assurer le caractère unique des nonces. Chaque utilisateur peut par exemple concaténer son identité avec un compteur mais, alors, un agresseur peut vérifier facilement le numéro des messages qui ont été envoyés pendant une session, et ensuite, le système doit gérer très soigneusement une base de données d'identités. Il est beaucoup plus satisfaisant, puisque dans ce cas il n'y a pas besoin d'une base de données, d'utiliser des nonces pseudo-aléatoires. Cependant, afin d'éviter les répétitions, le nombre de messages ne peut pas dépasser $\sqrt{2^n}$ quand R est un nonce de n bits.
- Une seconde complication est que les familles de fonctions que nous utilisons en pratique sont des permutations sur n bits. Il est alors probable que des étiquettes de n bits conduisent à une probabilité de collision égale à $q/2^n$ après q tentatives. Malheureusement, en raison du terme $\mathbf{Adv}_F^{\mathbf{prf}}(\mathcal{A})$, q ne peut pas pas dépasser $\sqrt{2^n}$. En d'autres termes, avec une famille $1/2^{128}$-quasi delta universelle de fonctions de hachage avec sorties 128 bits, nous devons utiliser la version 256 bits de l'algorithme de chiffrement bloc AES et des nonces aléatoires de taille 256 afin d'obtenir une sécurité qui ne dépasse pas celle obtenues avec des étiquettes de 128 bits. Pour ce niveau de sécurité, utiliser la version 128 bits d'un algorithme de chiffrement bloc est à l'évidence plus efficace.

Nous observons cependant que l'approche "randomisation" proposée pour la construction RMAC [68, 67] semble surmonter ces deux difficultés. Surtout elle conduit à des bornes de sécurité meilleures puisque au delà de la limite du paradoxe des anniversaires, même avec des nonces complètement aléatoires. Malheureusement, RMAC n'est pas un MAC efficace en logiciel. La construction FRMAC, le MAC dont nous évoquons l'analyse ici, essaie de combiner le meilleur des deux approches. Il consiste, comme dans l'approche de Wegman-Carter, en l'utilisation d'une fonction de hachage afin de réduire le message à une taille fixe et ensuite, comme dans RMAC, en l'application au résultat du hachage d'une permutation paramétrée. Bien que, pour les grands messages, les exécutions peuvent être identiques à celles de UMAC-2000. Pour les petits messages, on espère améliorer très sensiblement les performances car un algorithme de chiffrement avec une plus petite taille de bloc peut être utilisé.

Notations.
- $\mathcal{M}$ représente un message, c'est-à-dire une suite de bits. Sa taille est notée $|\mathcal{M}|$ ou m. La concaténation de deux messages $\mathcal{M}$ et $\mathcal{N}$ est indiqué par $\mathcal{M}||\mathcal{N}$.
- H représente une famille de 2^d fonctions de hachage. Une fonction dans la famille est indiquée par h. La sortie des fonctions de hachage est de taille n bits.
- F représente une famille de fonctions. Les éléments de la famille sont notées f. De la même manière, Π représente une famille de permutations et une permutation dans la famille est indiquée π.
- Soient (n, ν) deux entiers. L'ensemble $\mathrm{Rand}(n, \nu)$ représente l'ensemble de toutes les fonctions de $\{0,1\}^n$ vers $\{0,1\}^\nu$.
- Soit n un entier. L'ensemble $\mathrm{Perm}(n)$ représente l'ensemble de toutes les permutations de $\{0,1\}^n$.
- Soient (k, n) deux entiers. L'ensemble $\mathrm{FPerm}(k, n)$ représente l'ensemble de toutes les familles de 2^k permutations de $\{0,1\}^n$.

2.1 Fonctions ε-universelles

2.1.1 Définition

Dans la définition ci-dessous, ε_m est une fonction monotone croissante de $\mathbb{N}$ vers $[1/2^n, \infty[$.

Définition 1. *H est une famille ε_m-quasi universelle (ou ε_m-AU) de fonctions de hachage si, pour tout $\mathcal{M} \neq \mathcal{N}$ dans $\{0,1\}^m$, $|\{h \in H \mid h(\mathcal{M}) = h(\mathcal{N})\}| \leqslant \varepsilon_m 2^d$.*

Quand $\varepsilon_m = \varepsilon$ est une constante, H est une classique famille de fonctions de hachage ε-AU. Dans le contexte cryptographique, beaucoup de familles de fonctions de hachage ε-AU à sortie fixe sont ε_m-quasi universelles. Souvent, $\varepsilon_m = (cm + d)^e / 2^n$ avec c, d, e des constantes positives.

2.1.2 Construction

Comme remarqué par Bernstein à la fin de [13], beaucoup de familles de fonctions de hachage ε_m-AU efficaces peuvent être construites comme la réduction modulo des idéaux secrets dans des anneaux.

Plus précisément, soit $\mathcal{R}$ un anneau commutatif (avec élément neutre) et soit p, une fonction injective d'un ensemble de messages $\{0,1\}^*$ vers un sous-ensemble S de $\mathcal{R}$ tel que, pour n'importe quels messages $\mathcal{M}$ et $\mathcal{N}$, $p(\mathcal{M}) - p(\mathcal{N}) \in S$. Nous notons par S_m une famille de sous-ensembles finis de S tel que pour n'importe quels messages $\mathcal{M}$ et $\mathcal{N}$ dans $\{0,1\}^m$, $p(\mathcal{M}), p(\mathcal{N}), p(\mathcal{M}) - p(\mathcal{N}) \in S_m$ et tel que $S_1 \subset S_2 \subset \cdots \subset S$. De plus, soit $I = (\mathcal{I}_i)_{1,\ldots,2^d}$ de 2^d idéaux distincts de $\mathcal{R}$. À nos fins cryptographiques, ces idéaux doivent satisfaire deux conditions.

Cond. 1 : Pour n'importe quel élément $\mathcal{I}_i$ de I, l'anneau intègre $\mathcal{R}/\mathcal{I}_i$ est fini. Nous notons par 2^n la puissance de deux immédiatement plus grande que la cardinalité du plus grand anneau intègre et par q_i une fonction injective de $\mathcal{R}/\mathcal{I}_i$ vers $\{0,1\}^n$.

Cond. 2 : Il existe une fonction monotone croissante ε_m telle que

$$\forall z \in S_m,\ |\{\mathcal{I} \in I \mid z \bmod \mathcal{I} = 0\}| < \varepsilon_m 2^d \ .$$

Avec ces notations, nous considérons la famille H de fonctions de hachage $(h_i)_{1,\ldots,2^d}$ définie par,

$$h_i : \{0,1\}^* \longrightarrow \{0,1\}^n,\ \mathcal{M} \longrightarrow q_i(p(\mathcal{M}) \bmod \mathcal{I}_i) \ .$$

Lemme 1. *H est une famille de fonctions de hachage ε_m-AU.*

Preuve. Soient $\mathcal{M}$ et $\mathcal{N}$ deux messages distincts de $\{0,1\}^m$, alors

$$\begin{aligned}
|\{h \in H \mid h(\mathcal{M}) = h(\mathcal{N})\}| &= |\{i \mid q_i(p(\mathcal{M}) \bmod \mathcal{I}_i) = q_i(p(\mathcal{N}) \bmod \mathcal{I}_i)\}| \ , \\
&= |\{i \mid p(\mathcal{M}) - p(\mathcal{N}) \bmod \mathcal{I}_i = 0\}| \quad (q_i \text{ injective}) \ , \\
&\leqslant \max_{z \in S_m} |\{i \mid z \bmod \mathcal{I}_i = 0\}| \quad (p \text{ injective}) \ , \\
&\leqslant \varepsilon_m 2^d \quad (\text{Cond. 2}) \ .
\end{aligned}$$

□

Les conditions 1 et 2 peuvent sembler, au premier coup d'oeil, un petit peu obscures. Elles sont en fait les conséquences de propriétés mathématiques. La condition 1, par exemple, est vraie pour les idéaux maximaux dans les anneaux tel que leur groupe additif est un groupe abélien engendré de façon finie [44, Exercice 4.26, p.142], ce qui est généralement vrai pour les constructions de mathématique usuelles au-dessus de corps finis. La condition 2 est un corollaire du caractère unique et fini de la factorisation en idéaux pour une grande classe d'anneaux. C'est ainsi bien connu que dans le contexte des anneaux de Dedekind présent, par exemple, pour des corps globaux (i.e les corps de nombres ou les corps de fonction), nous avons factorisation unique en idéaux primordiaux [136]. Pour le cas plus général des anneaux noetherien (i. e. les algèbres de polynômes multivariés), nous avons un résultat similaire pour la décomposition primaire en idéaux maximaux [117].

2.1.3 Exemples

Soit $\mathbb{F}_q$ un corps fini et $\log_2 q = n$, nous donnons des exemples de telles constructions dans le tableau 1 et, pour chacun d'entre eux, une approximation $\tilde{\varepsilon}_m$ des bornes de collisions correspondantes.

TABLE 1 – Exemples de familles de fonctions de hachage ε_m-quasi universelles

$\mathcal{R}$	I	S_m	$p(\mathcal{M})$	$\tilde{\varepsilon}_m$
$\mathbb{Z}$	$\{(\pi) \mid \pi \text{ prime}, 2^{n-1} < \pi < 2^n\}$	$\{z \in \mathcal{R} \mid \lvert z\rvert \leqslant 2^m\}$	$\sum_{\lvert M_i\rvert=1} M_i 2^i$	$\frac{m}{2^n}$
$\mathbb{F}_2[X]$	$\{(\pi(X)) \mid \pi(X) \text{ irreducible}, \deg \pi(X) = n\}$	$\{z \in \mathcal{R} \mid \deg(z) \leqslant m\}$	$\sum_{\lvert M_i\rvert=1} M_i X^i$	$\frac{m}{2^n}$
$\mathbb{F}_q[X]$	$\{(X-\alpha) \mid \alpha \in \mathbb{F}_q\}$	$\{z \in \mathcal{R} \mid \deg(z) \leqslant \frac{m}{n}\}$	$\sum_{\lvert M_i\rvert=n} M_i X^i$	$\frac{m}{n\,2^n}$
$\mathbb{F}_q[X,Y]$	$\{(X-\alpha, Y-\beta) \mid \alpha,\beta \in \mathbb{F}_q\}$	$\left\{z \in \mathcal{R} \mid \deg(z) \leqslant \frac{m}{n}\right\}$	$\sum_{\lvert M_{ij}\rvert=n} M_{ij} X^i Y^j$	$\frac{\sqrt{m}}{\sqrt{n}\,2^n}$

2.2 Un MAC randomisé rapide

Par soucis d'exhaustivité, nous analysons la sécurité de la construction FRMAC pour n'importe quel famille de fonctions de hachage ε_m-AU.

2.2.1 Définition

Soit maintenant Π une famille de permutations de $\mathrm{Perm}(n)$, indexée par un bloc R de r bits. Un élément de cette famille est noté π_R. Le MAC que nous étudions est donné par

$$\mathrm{FRMAC}_\Pi(\mathcal{M}) = (\pi_R(h(\mathcal{M})), R) \quad , \tag{1}$$

où R est un bloc de r bits choisi au hasard pour la génération du MAC. La permutation π_R et la fonction h sont secrètes. Afin de vérifier un MAC $\mathcal{C} = (T, R)$ correspondant à un message $\mathcal{M}$, nous vérifions si $T = \pi_R(h(\mathcal{M}))$.

Soit H est une famille de fonctions de hachage ε_m-AU. Soit E un algorithme de chiffrement par bloc avec des blocs de taille n et des clefs de taille k, où $n \leqslant k$. Le MAC que nous considérons dans ce modèle est le suivant :

$$\mathrm{FRMAC}_E(\mathcal{M}) = (E_{K\oplus R}(h(\mathcal{M})), R) \ ,$$

où R est un bloc de n bits, choisi au hasard pour la génération du MAC. La valeur tirée au hasard R est de taille n. Si $k > n$, le bloc R est complété avec des zéros avant d'exécuter l'opération `XOR`.

2.2.2 Modèle de sécurité

- Adversaire Avant de donner le résultat de sécurité, nous définissons premièrement l'adversaire contre la construction FRMAC.

- Oracles Dans le cas général d'une construction FRMAC_Π, où Π est une famille quelconque de permutations, l'adversaire peut accéder aux machines de (dé)chiffrement par deux oracles.

- L'oracle de génération $\mathcal{O}_G$ choisit une valeur au hasard $R \in \{0,1\}^r$ et calcule le MAC $\mathcal{C} = (\pi_R(h(\mathcal{M})), R)$.
- L'oracle de vérification $\mathcal{O}_V$ vérifie si $\pi_R(h(\mathcal{M})) = T$ pour un triplet $(\mathcal{M}, T, R)$.

L'adversaire est autorisé à demander la génération de MACs différent pour un message seul. Dans ce cas, puisque l'adversaire peut toujours se débarrasser des duplications, nous supposons qu'un R différent est choisi chaque fois. Nous supposons aussi que l'adversaire ne demande jamais une question de vérification clairement invalide. Il ne demande ainsi jamais la vérification de $(\mathcal{M}, T', R)$ où $T \neq T'$ et où (T, R) est un MAC généré par $\mathcal{O}_G$ pour $\mathcal{M}$.

- Paramètres de l'adversaire Un adversaire est défini par trois paramètres. Le nombre L est la taille totale, en bits, des messages qui forment les requêtes aux deux oracles. Le nombre τ mesure le temps d'exécution maximal de l'adversaire. Le nombre μ représente une limite sur la taille de code de l'adversaire.

- Buts de l'adversaire Le but de l'adversaire est de forger avec succès un MAC valide pour un message $\mathcal{M}$. Quand ceci arrive, nous considérons que l'adversaire a gagné. Un tel événement arrive quand l'oracle de vérification retourne "MAC valide" sur une demande $(\mathcal{M}, T, R)$ où (T, R) n'est pas une sortie de l'oracle de génération pour $\mathcal{M}$.

La probabilité de succès de l'adversaire $\mathcal{A}$ est noté $\mathbf{Adv}^{\text{forge}}_{\text{FRMAC}}(\mathcal{A})$. Comme expliqué au-dessus, c'est sa probabilité de recevoir la réponse "MAC valide" avec accès aux deux oracles $\mathcal{O}_G$ et $\mathcal{O}_V$:

$$\mathbf{Adv}^{\text{forge}}_{\text{FRMAC}}(\mathcal{A}) = \mathbf{Pr}[\mathcal{A}^{\mathcal{O}_G, \mathcal{O}_V} \text{ obtient la réponse "MAC valide"}] \ .$$

Nous surchargeons cette notation et nous écrivons $\mathbf{Adv}^{\text{forge}}_{\text{FRMAC}}(\tau, \mu, L)$ pour la valeur maximale de l'avantage précédent sur tous les adversaires de paramètres (τ, μ, L).

2.2.3 Sécurité dans le modèle de la théorie de l'information

L'analyse de sécurité nécessite deux résultats plus précis pour les familles de fonctions de hachage ε_m-AU.

Lemme 2 (Collision dans un groupe de messages)**.** *Soient $\mathcal{M}_1, \ldots, \mathcal{M}_q$ un groupe de q messages distincts de tailles $m_1, \ldots m_q$ bits, alors*

$$\mathbf{Pr}[\exists i \neq j \text{ such that } h(\mathcal{M}_i) = h(\mathcal{M}_j) \mid h \xleftarrow{R} H] \leqslant q \sum_{i=1}^{q} \varepsilon_{m_i} \ .$$

Démonstration. Puisqu'il est toujours possible de numéroter les messages de façon différente, nous pouvons supposer que les tailles m_i sont ordonnées de façon décroissante. Soit $P = \mathbf{Pr}[\exists i \neq j \text{ tel que } h(\mathcal{M}_i) = h(\mathcal{M}_j) \mid h \xleftarrow{R} H]$, alors

$$P \leqslant \sum_{i=1}^{q} \sum_{j>i} \mathbf{Pr}[h(\mathcal{M}_j) = h(\mathcal{M}_i) \mid h \xleftarrow{R} H] \leqslant \sum_{i=1}^{q} \sum_{j>i} \varepsilon_{\max(m_j, m_i)} \leqslant \ q \sum_{i=1}^{q} \varepsilon_{m_i} \quad \text{(Def. 1)}.$$

□ □

Lemme 3 (Collision avec un message de référence)**.** *Soient* $\mathcal{M}_0$, $\mathcal{M}_1$, …, $\mathcal{M}_q$ *un groupe de* $q+1$ *messages distincts de tailles* m_0, m_1, …, m_q *bits, alors*

$$\mathbf{Pr}[\exists i \in \{1,\ldots,q\} \text{ tel que } h(\mathcal{M}_i) = h(\mathcal{M}_0) \mid h \xleftarrow{R} H] \leqslant q \max_{0\leqslant i\leqslant q} \varepsilon_{m_i} \ .$$

Démonstration. Soit $P = \mathbf{Pr}[\exists i \in \{1,\ldots,q\} \text{ tel que } h(\mathcal{M}_i) = h(\mathcal{M}_0) \mid h \xleftarrow{R} H]$, alors

$$P \leqslant \sum_{i=1}^{q} \mathbf{Pr}[h(\mathcal{M}_i) = h(\mathcal{M}_0) \mid h \xleftarrow{R} H] \leqslant \sum_{i=1}^{q} \varepsilon_{\max(m_0,m_i)} \leqslant q \max_{0\leqslant i\leqslant q} \varepsilon_{m_i} \quad \text{(Def. 1)} \ .$$

□ □

Soit FRMAC_Π la construction définie au paragraphe 2.2 par l'équation (1). Nous étudions, en premier, sa sécurité dans le modèle de la théorie de l'information. C'est-à-dire que nous supposons que l'adversaire a une puissance calculatoire illimité et l'on quantifie ses chances de succès en fonction de la taille totale en bits de ses requêtes aux oracles.

On peut alors montrer que la sécurité de FRMAC_Π dans le modèle de la théorie de l'information est donné par le théorème 2 [69].

Théorème 2. *Soit* Π *une famille de* 2^r *permutations aléatoires de* $\text{Perm}(n)$*. Si* $r = n$ *et si* $n \geqslant 3$*, nous avons* $\Sigma_L = \dfrac{1}{L} \max_{\substack{\forall \mathcal{M}_i \text{ s.t.} \\ \sum |\mathcal{M}_i| = L}} \sum_i \varepsilon_{m_i}$,

$$\mathbf{Adv}^{\text{forge}}_{\text{FRMAC}_\Pi}(\tau,\mu,L) \leqslant L\left(\frac{n+1}{2^n} + n\Sigma_L + \varepsilon_L\right) + \frac{1}{2^n} \ .$$

Il n'est pas difficile de remarquer que $\Sigma_L \leqslant \varepsilon_L$. Le théorème 2 conduit donc à

$$\mathbf{Adv}^{\text{forge}}_{\text{FRMAC}_\Pi}(\tau,\mu,L) \leqslant L\,(n+1)\left(\varepsilon_L + \frac{1}{2^n}\right) + \frac{1}{2^n} \ . \tag{2}$$

2.2.4 Sécurité dans le modèle à clefs corrélées

Dans l'étude dans le modèle standard pour un algorithme de chiffrement par bloc E du schéma FRMAC_E , l'adversaire n'a plus une puissance de calcul illimitée illimité mais il peut aussi gagner en distinguant l'algorithme de chiffrement par bloc E d'une famille de permutations tirée au hasard.

Puisque l'adversaire a accès à un algorithme de chiffrement par bloc avec des clefs distinctes, nous utilisons le modèle de sécurité proposé par Bellare et Kohno dans [11]. Ce modèle donne des preuves dans le modèle standard dans le cas des attaques à clefs corrélées.

$$\mathbf{Adv}^{\textbf{prp-rka}}_{\texttt{XOR}_r,E}(\mathcal{A}) = \Big|\mathbf{Pr}[K \xleftarrow{R} \{0,1\}^k \ : \ \mathcal{A}^{E_{\cdot\oplus K}}(\cdot) = 1] - \mathbf{Pr}[K \xleftarrow{R} \{0,1\}^k, \Pi \xleftarrow{R} \text{FPerm}(k,n) \ : \ \mathcal{A}^{\Pi_{\cdot\oplus K}}(\cdot) = 1]\Big| \ .$$

Le modèle d'attaque permet à l'adversaire de choisir un nonce R de r bits et d'obtenir alors la valeur de l'algorithme de chiffrement par bloc, sur une donnée de son choix, sous la clef $R \oplus K$ (si $r < k$, R est complété avec des zéros pour cette opération). Cet avantage mesure le succès de l'adversaire pour déterminer si l'oracle utilise l'algorithme de chiffrement par bloc E ou une famille de permutations aléatoires.

Comme avant, nous surchargeons cette notation pour tous les adversaires de paramètres τ, μ et L. Le théorème 3 donne la sécurité de la construction dans le modèle calculatoire.

Théorème 3. *Soit E un algorithme de chiffrement par bloc avec des blocs de taille n et des clefs de taille k, avec $n \leqslant k$. Si $n \geqslant 3$, nous avons,*

$$\mathbf{Adv}^{\text{forge}}_{\text{FRMAC}_E}(\tau, \mu, L) \leqslant L\,(n+1)\left(\frac{1}{2^n} + \varepsilon_L\right) + \frac{1}{2^n} + \mathbf{Adv}^{\textbf{prp-rka}}_{\texttt{XOR}_n, E}(\tau, \mu, L)\ . \tag{3}$$

La preuve de ce théorème dérive immédiatement du théorème 2 et de la définition de l'avantage $\mathbf{Adv}^{\textbf{prp-rka}}_{\texttt{XOR}_n, E}(\tau, \mu, L)$. En effet, si un adversaire peut forger un MAC valide contre FRMAC_E, il est soit capable de forger directement contre FRMAC_Π, soit il fournie une façon de distinguer entre les deux situations.

Les algorithmes de chiffrement par bloc les plus modernes, y compris l'AES, sont conçus avec le but de résister aux attaques à clefs corrélées de type $\texttt{XOR}_n$. Cela est d'autant plus vraie, pour les algorithmes de chiffrement par bloc avec des clefs de taille k deux fois plus grande que leur bloc de taille n, il semble raisonnable de supposer que $\mathbf{Adv}^{\textbf{prp-rka}}_{\texttt{XOR}_n, E}(\tau, \mu, L) \simeq \mathbf{Adv}^{\textbf{prp}}_{E}(\tau, \mu, L) \simeq L/2^n$.

2.2.5 Conclusion

Le code d'authentification de message FRMAC est une construction de type RMAC pour laquelle une famille de fonctions de hachage ε-quasi universelle est utilisée au lieu d'une chaîne CBC. Ce schéma permet ainsi l'utilisation de nonces tirés au hasard. Quand les étiquettes et les nonces sont de taille n bits, nous en tirons comme conséquence que la borne de sécurité de FRMAC dans le modèle de la théorie de l'information (2) est seulement $n+1$ fois plus grande que la borne classique pour les MACS de type Wegman-Carter. De plus, puisque FRMAC profite de la résistance aux attaques à clefs corrélées de la plupart des algorithmes de chiffrement par bloc, la borne de sécurité dans le modèle calculatoire (3) n'est plus pénalisée par le lemme classique de transition PRF/PRP. Ceci produit finalement une borne de sécurité qui est, comme dans RMAC, au delà du paradoxe des anniversaires.

En retour, l'utilisation de ce MAC dans les vraies applications a tendance à être plus facile. Utilisé avec une fonction de hachage efficace en logiciel, ce MAC est d'ailleurs l'un des plus rapide connu aujourd'hui.

Logarithmes discrets dans les corps finis

— *Le crible algébrique*

Soit G un groupe fini cyclique, alors il existe un élément g de G tel que pour tout élément non-nul y de G, il existe un entier x tel que $y = g^x$. L'entier x est classiquement appelé l'index ou le logarithme discret de y en base g. Il est souvent noté $\log_g(y)$. De façon analogue à la fonction logarithme népérien, la fonction $\log_g$ satisfait $\log_g yz = \log_g y + \log_g z$ modulo le cardinal de G et une application immédiate permet de ramener le calcul de la loi de groupe à des sommes d'entiers.

Ainsi, calculer le produit de deux éléments du groupe multiplicatif d'un corps fini par cette méthode est connue sous le nom de méthode de Zech (cf. chapitre précédent). Bien que séduisante, il est apparu très vite que cette méthode de multiplication était inappropriée pour des corps finis de grande taille, c'est-à-dire ceux pour lesquels on ne peut pas précalculer par avance une table. En fait, si calculer y à partir de x est facile (méthode d'exponentiation binaire), l'inverse est considéré comme difficile. Aujourd'hui les algorithmes les plus efficaces de résolution, ceux que nous étudions dans ce chapitre, sont de complexités sous-exponentielles en la taille du corps. À l'opposé, il est bien connu que la cryptographie a su tirer partie de cette difficulté. Une variante du problème du logarithme discret, le problème Diffie-Hellman, est ainsi à la base de nombreux mécanismes à clef publique [43].

L'amélioration des algorithmes de calculs de logarithmes discrets dans les corps finis est motivée par ses impacts en cryptographie. Elle permet bien sûr de justifier le dimensionnement nécessaire aux algorithmes asymétriques dont la sécurité en dépend. De façon annexe, elle peut aussi être un outil d'évaluation d'autres mécanismes cryptographiques, par exemple, l'étude des registres linéaires à la base des générateurs pseudo-aléatoires [115].

Selon Shoup [158], les meilleurs algorithmes possibles pour calculer des logarithmes dans un groupe arbitraire G de cardinalité q est de complexité en temps au mieux $O(q^{\frac{1}{2}})$. Un algorithme atteignant ces bornes pour $\mathbb{F}_{p^n}$ peut être le suivant :

1. Grâce à l'algorithme Pohlig-Hellman, le calcul de x est équivalent au calcul de logarithmes discrets dans les groupes cycliques dont l'ordre q sont des diviseurs de $p^n - 1$.
2. Calculer des logarithmes discrets dans un groupe cyclique de $\mathbb{F}_{p^n}^*$ peut être fait en $O(q^{\frac{1}{2}})$ grâce à la méthode pas de bébés, pas de géants, ou mieux, grâce à des méthodes de type Pollard-ρ.

Bien qu'efficaces pour des corps finis dont l'ordre est divisible par des nombres premiers de petites tailles, ces méthodes demeurent exponentielles en la taille du plus grand diviseur premier de $p^n - 1$. C'est pourquoi, nous nous intéressons ici à des méthodes plus spécifiques, les méthodes dites de type "index-calculus" [118]. Nous commençons par en rappeler les principes (cf. paragraphe 1). Puis nous présentons les améliorations que nous avons apportées en collaboration avec Joux [78, 79]. Ces dernières concernent le crible algébrique général pour les corps finis de grande caractéristique (cf. paragraphe 2) et le crible dans les corps de fonctions pour les corps finis de petite caractéristique (cf. paragraphe 3). On améliore le choix des corps de nombres ou de fonctions. On explique aussi comment, après une phase de précalcul qui se trouve être la difficulté majeure, on peut ensuite dans les deux cas calculer rapidement le logarithme discret d'un élément arbitraire.

1 Les algorithmes “Index calculus”

1.1 Le principe

En toute généralité, une description générique des méthodes index-calculus pour calculer le logarithme discret en base un élément g de $\mathbb{F}_{p^n}$ peut-être faite comme suit.

Étape 1 : on fixe un sous-ensemble $S = \{\gamma_1, \ldots, \gamma_{|S|}\}$ de $< g >$ appelé la “base de lissité” et on cherche des relations faisant intervenir les éléments de S de la forme $\prod_{(\epsilon,\gamma)\in\mathbb{Z}\times S} \gamma^\epsilon = 1$, ce qui conduit à des équations du type

$$\sum_{(\epsilon,\gamma)\in\mathbb{Z}\times S} \epsilon \log_g \gamma = 0 \bmod \ell \tag{1}$$

où $\# < g >= \ell$.

Étape 2 : quand on a suffisamment de telles relations, on calcule des résidus $\log_g \gamma$ en inversant modulo ℓ le système linéaire correspondant.

Étape 3 : pour calculer le logarithme discret d'un élément quelconque y de g, on essaie des entiers aléatoires ν jusqu'à ce que $g^\nu y$ s'écrive comme un produit d'éléments de S, c'est-à-dire $g^\nu y = \prod_{(\epsilon,\gamma)\in\mathbb{Z}\times S} \gamma^\epsilon$. Alors,

$$\log_g y = (-\nu + \sum_{(\epsilon,\gamma)\in\mathbb{Z}\times S} \epsilon \log_g \gamma) \bmod \ell.$$

Dans les faits, l'efficacité de la méthode, et en particulier les complexités en temps associées, dépendent de la stratégie utilisée pour le choix de la base S et de la méthode pour rechercher des relations.

1.2 Analyse de complexité d'un cas d'école

Nous nous plaçons ici dans $\mathbb{F}_p$ avec $\ell = (p-1)/2$, un nombre premier. Étant donné $\gamma \in \mathbb{Z}/p\mathbb{Z}$, on définit aussi $\gamma^\uparrow$ comme étant le plus petit entier strictement positif de la classe de congruence γ modulo p. Un choix naturel pour S est formé du générateur g et de l'ensemble des nombres premiers q inférieurs à une borne B. Si la factorisation sur $\mathbb{Z}$ d'entiers de la forme $(g^\nu)^\uparrow$ pour des entiers ν choisis aléatoirement ne fait intervenir que des nombres premiers q de S, on en déduit alors une relation du type (1).

Il est dans ce cas, une fois rappelés quelques résultats sur la densité des nombres friables, relativement aisé de retrouver une complexité sous-exponentielle.

1.2.1 Rappels sur les fonctions L

Étant donné des réels π, ν et $\lambda \neq 0$, définissons pour π tendant vers l'infini, la quantité

$$L_\pi[\nu, \lambda] = e^{\lambda(\log \pi)^\nu (\log\log \pi)^{1-\nu}} .$$

Par la suite, on utilisera ces fonctions pour des estimations asymptotiques de complexité et classiquement, on abrège $L_\pi[\nu, \lambda]$ pour $L_\pi[\nu, \lambda + o(1)]$ ou encore $L_\pi[\nu]$ pour$L_\pi[\nu, \lambda]$. Elles vérifient,

$$\begin{aligned} L_\pi[\nu_1]L_\pi[\nu_2] &= L_\pi[\max(\nu_1, \nu_2)] \text{ si } \nu_1 \neq \nu_2 \text{ et} \\ L_\pi[\nu, \lambda_1]L_\pi[\nu, \lambda_2] &= L_\pi[\nu, \lambda_1 + \lambda_2]. \end{aligned}$$

De même, on a le résultat suivant.

Théorème 4. *Étant donné des constantes ν, λ, ω et μ telles que $1 > \nu > \omega > 0$ et $\lambda, \mu > 0$, la probabilité qu'un entier de taille $L_\pi[\nu, \lambda]$ soit $L_\pi[\omega, \mu]$-lisse est lorsque π tend vers l'infini asymptotiquement égale à*

$$L_\pi[\nu - \omega, -\lambda(\nu - \omega)/\mu].$$

1.2.2 Complexité

Supposons en toute première analyse que $B = L_p[\alpha]$ pour $0 \leq \alpha \leq 1$. On cherche des entiers de taille $L_p[1]$ qui soient B-lisse. La probabilité d'occurrence de telles relations est $L_p[1-\alpha]$, il en faut $L_p[\alpha]$ et donc, le temps de la première étape est $L_p[\max(1-\alpha, \alpha)]$.

Le temps de la seconde étape, celle de l'inversion modulaire, est polynomial en la taille de S, c'est-à-dire égal à $L_p[\alpha]$.

Enfin, dans la dernière étape, on recherche un entier B-lisse de taille $L_p[1]$, ce qui est réalisable en temps $L_p[1-\alpha]$.

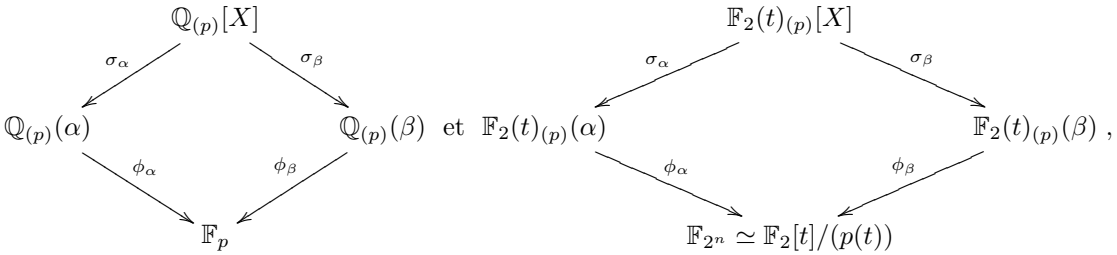
Le temps total est ainsi $L_p[\max(\alpha, 1-\alpha)]$. Il vaut donc au minimum $L_p[1/2]$, réalisé pour $\alpha = 1/2$.

Remarque 2. *Une analyse attentive permet de s'assurer que la complexité de cet algorithme probabiliste ne repose sur aucune heuristique. Ce n'est malheureusement plus le cas des algorithmes qui suivent, principalement par ce que l'on cherche à y factoriser des entiers ou des polynômes de tailles plus petites, qui ne sont plus complètement aléatoires.*

1.3 Décomposition en idéaux dans les corps globaux

Considérer la lissité d'objets directement dans $\mathbb{F}_{p^n}$ est inapproprié, simplement parce que tout élément y est une unité. C'est pourquoi, déjà dans la version naïve considérée dans le paragraphe précédent, on se plaçait via l'opération % dans $\mathbb{Z}$. Les algorithmes les plus modernes de calcul de logarithme discrets contournent la difficulté en obtenant des bases de lissité par réduction modulo des idéaux premiers dans l'anneau des entiers d'un corps global.

En fait, les deux cas importants dans les applications, ceux que nous considérons dans ce chapitre, peuvent être résumés, respectivement pour $\mathbb{F}_p$ et $\mathbb{F}_{2^n}$, par les diagrammes commutatifs



où α et β sont des entiers algébriques dont les polynômes minimaux ont une racine commune μ modulo p (resp. des fonctions dont les polynômes minimaux ont une racine commune $\mu(t)$ modulo $p(t)$) et où les flèches sont définies par $\sigma_\alpha : X \longrightarrow \alpha$, $\sigma_\beta : X \longrightarrow \beta$, $\phi_\alpha : \alpha \longrightarrow \mu$ et $\phi_\beta : \beta \longrightarrow \mu$.

Tout comme dans $\mathbb{F}_{p^n}$, considérer la lissité d'objets dans les anneaux $\mathbb{Q}_{(p)}[X]$ où $\mathbb{F}_2(t)_{(p)}[X]$ conduit à une impasse car la majeure partie des éléments y sont irréductibles. Par contre, ce n'est plus le cas au niveau intermédiaire, celui des corps de nombres ou de fonctions. C'est pourquoi les bases de lissité S qui sont considérées par la suite proviennent d'idéaux premiers de normes bornées dans l'anneau des entiers de ces corps.

Schématiquement, ces algorithmes recherchent des couples (a, b) de rationnels ou de polynômes de $\mathbb{F}_2[t]$ tels que les idéaux principaux de générateurs $a + b\alpha$ et $a + b\beta$ se factorisent simultanément sur l'ensemble des idéaux premiers choisis pour la base de lissité. Lorsque l'une de ces relations est découverte, il est possible de les réduire dans le corps fini pour obtenir une égalité entre les deux "factorisations réduites".

À ce stade plusieurs questions se posent et sont l'objet des paragraphes suivants. Les principales sont les suivantes.

- D'un point de vue théorique, comment lever les obstructions liées à l'utilisation d'anneaux non-factoriels avec des unités non-triviales ?
- D'un point de vue algorithmique, quels sont les meilleurs choix pour les corps de nombres ou de fonctions et, une fois ces choix faits, comment organiser efficacement les calculs ?

2 Logarithmes discrets dans $\mathbb{F}_p$

Il existe en toute généralité trois familles principales d'algorithmes pour calculer des logarithmes discrets avec des algorithmes de type "index calculus" dans $\mathbb{F}_p$.

- La méthode originale due à Kraitchik [118] dont la complexité est égale à $L_p[\frac{1}{2}, c]$.
- La méthode des entiers de Gauss due à Coppersmith, Odlyzko et Schroeppel [35] dont la complexité est égale à $L_p[\frac{1}{2}, 1]$.
- Le crible algébrique général tel qu'amélioré par Schirokauer [149] suite à une proposition de Gordon [58] dont la complexité est égale à $L_p[\frac{1}{3}, (\frac{64}{9})^{\frac{1}{3}}]$.

Afin de faciliter la compréhension, nous introduisons d'abord la méthode des entiers de Gauss avec le formalisme qui sera nécessaire à la présentation dans un second temps du crible algébrique. Nous terminons par une comparaison des deux méthodes au travers d'expérimentations que nous avons menées.

2.1 La méthode des entiers de Gauss

La méthode dite des entiers de Gauss utilise comme corps de nombres $\mathbb{Q}$ et un corps de nombres quadratique imaginaire. On choisit un petit entier r tel que modulo p, $-r$ soit égal au carré d'un élément μ. On écrit par ailleurs $\mu = n/d$ dans $\mathbb{F}_p$ avec $n, d \simeq O(\sqrt{p})$ et on considère les corps de nombres $\mathbb{Q}(\alpha) \simeq \mathbb{Q}$ et $\mathbb{Q}(\beta)$ respectivement définis par

$$f_\alpha(X) = dX - n \text{ et } f_\beta(X) = X^2 + r$$

dont on note les anneaux d'entiers $\mathcal{O}_\alpha \simeq \mathbb{Z}$ et $\mathcal{O}_\beta$.

Il est alors facile de donner dans $\mathbb{Q}(\alpha)$ un sens en terme d'entiers algébriques à une factorisation en idéaux premiers et d'envoyer ces entiers dans $\mathbb{F}_p$ par, $(a + b\alpha) \to a + b\mu$. Pour les idéaux de $\mathcal{O}_\beta$, la situation est à peine plus difficile lorsque le groupe des classes de $\mathcal{O}_\beta$ est de cardinalité un puisqu'alors, pour tout idéal $\mathfrak{p}$ de O_β,

$$\exists (a_\mathfrak{p}, b_\mathfrak{p}) \in \mathbb{Z}^2 \text{ tel que } \mathfrak{p} = (a_\mathfrak{p} + b_\mathfrak{p}\beta).$$

L'application de réduction est alors simplement donnée par $a_\mathfrak{p} + b_\mathfrak{p}\beta \to a_\mathfrak{p} + b_\mathfrak{p}\mu$. Grâce à ces définitions, on peut obtenir la base de lissité à partir de

$$\mathcal{S} = \{p \in \mathbb{Z}, p \text{ nombre premier } < B_\alpha\} \cup \{a_\mathfrak{p} + b_\mathfrak{p}\mu, \mathfrak{p} \text{ idéal premier de } O_\beta, \mathrm{Norm}(\mathfrak{p}) < B_\beta\} \cup \{\text{générateurs du groupe des unités}\},$$

où B_α, B_β sont des bornes qui conduisent à des probabilités de lissité suffisamment hautes. L'étape 1 de la méthode générique est alors simplement reformulée comme suit.

Étape 1 : on recherche des couples (a, b) tels que $(a + b\alpha)$ et $(a + b\beta)$ soient simultanément lisses. Alors

$$v \prod_{p \in \mathcal{S}} p^{e_p} = a + b\mu = u \prod_{\mathfrak{p} \in \mathcal{S}} (a_\mathfrak{p} + b_\mathfrak{p}\mu)^{e_\mathfrak{p}} \text{ où } u \text{ et } v \text{ sont des unités.}$$

Les calculs majeurs réalisés avec cette méthode sont donnés dans le tableau 1.

Taille de p	Quand	Qui	Commentaires
58 chiffres	1991	B.A. Lamacchia et A.M. Odlyzko [95]	2 mois sur un DEC VAX 8850
85 chiffres	1996	D. Weber [171]	120 stations de travail $\simeq$ 30 MipsYear
90 chiffres	1998	A. Joux et R. Lercier [72]	5 mois sur un PC 180 MHz

TABLE 1 – Calculs de logarithme discret avec la méthode des entiers de Gauss

2.2 Le crible algébrique

Le crible algébrique est une généralisation de la méthode des entiers de Gauss à des corps de nombres arbitraires $\mathbb{Q}(\alpha)$ ou $\mathbb{Q}(\beta)$. On a toujours, en corollaire du théorème de Dedekind, les factorisations

$$(a+b\alpha) = \prod_{\mathfrak{p}\in\mathcal{S}} \mathfrak{p}^{e_\mathfrak{p}} \text{ et } (a+b\beta) = \prod_{\mathfrak{p}\in\mathcal{S}} \mathfrak{p}^{e_\mathfrak{p}}.$$

Le problème est que l'on ne peut plus les réduire aussi facilement dans $\mathbb{F}_p$. Les obstructions, (triviales pour $\mathbb{Q}$) proviennent du groupe des classes et des unités de ces corps qui, pour certains des corps considérés en pratique, ne peuvent pas être calculées. On doit à Schirokauer une méthode permettant de lever cette difficulté. Les plus gros calculs réalisés à ce jour reposent sur ces idées (cf. tableau 2).

Taille de p	Quand	Qui	Commentaires
25 chiffres	1994	D. Weber [170]	1 jour sur une sparc ELC
65 chiffres	1995	D. Weber [170]	130 stations de travail $\simeq$ 5 MipsYear
85 chiffres	1998	D. Weber [171]	130 stations de travail $\simeq$ 44.5 MipsYear
100 chiffres	1999	A. Joux et R. Lercier [73]	1 année sur un PC 450 MHz
110 chiffres	2000	A. Joux et R. Lercier [77]	1 mois sur 4 processeurs alpha 525 MHz
120 chiffres	2001	A. Joux et R. Lercier [76]	2 mois sur 4 processeurs alpha 525 MHz

TABLE 2 – Calculs de logarithme discret avec le crible algébrique général

Nous passons maintenant en revue les grandes étapes d'un crible algébrique général :
- trouver de bons corps de nombres (cf. paragraphe 2.2.1),
- cribler efficacement pour trouver des relations (cf. paragraphe 2.2.2),
- inverser le système linéaire (cf. paragraphe 2.2.4),
- résoudre des instances particulières du logarithme discret (cf. paragraphe 2.2.5).

2.2.1 Détermination des polynômes

Soient $f_\alpha(X)$ et $f_\beta(X)$ les polynômes de définition des entiers algébriques α et β. Ces polynômes doivent satisfaire les conditions suivantes :
- $f_\alpha(X)$ et $f_\beta(X)$ irréductibles sur $\mathbb{Z}$,
- les coefficients de $f_\alpha(X)$ et $f_\beta(X)$ sont premiers entre eux,
- $f_\alpha(X) \neq \pm f_\beta(X)$,
- $\exists \mu \in \mathbb{F}_p^*,\ f_\alpha(\mu) = f_\beta(\mu) = 0$.

La lissité des idéaux $(a+b\alpha)$ et $(a+b\beta)$ étant d'autant meilleure que leurs normes sont petites, il est de plus naturel de rechercher des polynômes $f_\alpha(X)$ et $f_\beta(X)$ avec des coefficients aussi petits que possible. Cependant, le fait d'avoir une racine commune μ dans $\mathbb{F}_p$ implique que p doit diviser le résultant de $f_\alpha(X)$ et $f_\beta(X)$).

- Les méthodes connues. Trois méthodes étaient, avant nos calculs, connues pour trouver des polynômes $f_\alpha(X)$ et $f_\beta(X)$ vérifiant les conditions précédentes.

La méthode des entiers de Gauss. Elle conduit à

$$f_\alpha(X) = a_1 X + a_0 \text{ et } f_\beta(X) = X^2 + r,$$

avec les coefficients a_i de l'ordre de $p^{\frac{1}{2}}$ et $r = O(1)$.

La méthode de Montgomery. Elle conduit à

$$f_\alpha(X) = a_2X^2 + a_1X + a_0 \text{ et } f_\beta(X) = b_2X^2 + b_1X + b_0,$$

avec les coefficients a_i et b_i de l'ordre de $p^{\frac{1}{4}}$.

La décomposition en base μ. Elle conduit à

$$f_\alpha(X) = X - \mu \text{ et } f_\beta(X) = \sum_{i=0}^{d} b_iX^i,$$

avec μ et les coefficients b_i de l'ordre de $p^{\frac{1}{d+1}}$ et $\sum_{i=0}^{d} b_i\mu^i = p$.

- La méthode Joux-Lercier. La méthode que nous décrivons maintenant conduit à des normes plus petites que celles obtenues avec les méthodes précédentes [79]. Dans un premier temps, on choisit un polynôme $f_\beta(X)$ de degré $d+1$ avec des coefficients aussi petits que possible ayant une racine μ dans $\mathbb{F}_p$. On applique alors l'algorithme LLL au réseau

$$\begin{pmatrix} K & K\mu & K(\mu^2 \bmod p) & \cdots & K(\mu^d \bmod p) \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

où K est une grande constante arbitraire. On obtient alors un vecteur $(0, a_0, \dots a_d)^t$ tel que les coefficients a_i soient de l'ordre de $p^{\frac{1}{d+1}}$. Ainsi, puisque $\sum_{i=0}^{d} a_i\mu^i = 0$, on peut poser

$$f_\alpha(X) = \sum_{i=0}^{d} a_iX^i.$$

2.2.2 Obtention des relations avec les idées de Schirokauer

Soient S_α et S_β un ensemble d'idéaux premiers de degré un et de petites normes dans $\mathcal{O}_\alpha$ et $\mathcal{O}_\beta$. Alors les techniques de crible que nous décrivons plus loin conduisent à $|S_\alpha| + |S_\beta| + O(1)$ couples d'équations de la forme

$$(a + b\alpha) = \prod_{\mathfrak{p}\in S_\alpha} \mathfrak{p}^{e_\mathfrak{p}} \text{ et } (a + b\beta) = \prod_{\mathfrak{p}\in S_\beta} \mathfrak{p}^{e_\mathfrak{p}}.$$

Grâce alors a une inversion matricielle réalisée modulo $p-1$, nous avons,

$$\forall \mathfrak{p} \in S_\alpha,\ \mathfrak{p} = I_\mathfrak{p}^{p-1} \prod_{(a,b)} (a + b\alpha)^{e_{a,b}} \text{ et } \forall \mathfrak{p} \in S_\beta,\ \mathfrak{p} = I_\mathfrak{p}^{p-1} \prod_{(a,b)} (a + b\beta)^{e_{a,b}}.$$

Il reste de plus $|S_\beta| + O(1)$, $|S_\alpha| + O(1)$ équations supplémentaires de la forme

$$I_\alpha^{p-1} = \prod_{(a,b)} (a + b\alpha)^{e_{a,b}} \text{ et } I_\beta^{p-1} = \prod_{(a,b)} (a + b\beta)^{e_{a,b}}$$

(ou les idéaux I_α, I_β et $I_\mathfrak{p}$ sont non-principaux). Sous de bonnes conditions sur $p-1$, en particulier l'absence de puissances parmi ses facteurs, on peut pour chaque équation,

$$I_\alpha^{p-1} = \prod_{(a,b)} (a + b\alpha)^{e_{a,b}},$$

calculer, avec (r_1, r_2) la signature de $Q(\alpha)$, $r_1 + r_2 - 1$ quantités $(\lambda_1, \ldots, \lambda_{r_1+r_2-1})$ appelées "maps" de Schirokauer. Nous référons à [149] pour une définition précise de ces "maps", notons simplement qu'elles sont analogues à des logarithmes q-adiques de $(a + b\alpha)$ pour les facteurs premiers q de $p - 1$. Via une algèbre linéaire faite modulo $(p-1)$, il est alors facile de calculer une combinaison adéquate des dernières $r_1 + r_2 - 1$ équations avec chacune des premières $|S_\beta|$ équations pour obtenir $|S_\beta|$ nouvelles équations telles que leur vecteur de "maps" est nul.

Alors, Schirokauer explique que si la conjecture de Leopoldt est vraie, ces équations ne font intervenir que des entiers algébriques, i.e,

$$\exists \delta \in \mathcal{O}_\alpha,\ \delta^{p-1} = \prod_{(a,b)} (a + b\alpha)^{e_{a,b}}.$$

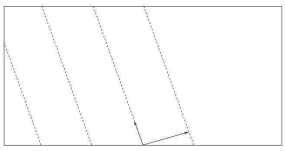
Remarque 3. *Pour faciliter l'exposition, nous avons décomposé ici l'algèbre linéaire en quelques sous-étapes. En "grandeur nature", il est bien entendu que nous la réalisons en une seule fois.*

2.2.3 Le crible par vecteurs

Avec un crible, plutôt que de tester successivement la lissité de chacun des idéaux principaux $(a+b\alpha)$ ou $(a + b\beta)$ avec les idéaux de la base de lissité, on procède à l'inverse. On marque dans le tableau d'abscisses a et d'ordonnées b, les idéaux principaux multiples des idéaux premiers de la base de lissité. En fait, puisque les idéaux premiers que nous considérons sont de degré un, il existe une racine ρ de $f_\alpha(X) \bmod \pi$ où $\pi = \mathrm{Norm}(\mathfrak{p})$ et les entiers multiples de $\mathfrak{p}$ appartiennent au réseau $L_\mathfrak{p}$ défini par

$$\begin{pmatrix} \rho & \pi \\ 1 & 0 \end{pmatrix}.$$

Lorsque l'on crible sur de grands tableaux, ce qui est le cas en pratique, on préfère généralement travailler dans le sous-réseau défini par un idéal $\mathfrak{q}$. Cette technique s'appelle un crible avec "special-q". Précisément, on considère les entiers algébriques $a + b\alpha$ qui sont multiples d'un idéal premier $\mathfrak{q}$ de norme moyenne. De tels entiers algébriques appartiennent à un réseau $L_\mathfrak{q}$. Comme par ailleurs tout entier algébrique multiple d'un idéal premier $\mathfrak{p}$ de la base appartient par définition à un réseau $L_\mathfrak{p}$, on est finalement rammené à énumérer les points du réseau $L_\mathfrak{q} \cap L_\mathfrak{p}$ dont il est aisé de déterminer une base.



Remarque 4. *En pratique, il est important d'utiliser comme base de ces réseaux des vecteurs courts pour accélérer le crible.*

2.2.4 L'algèbre linéaire

À la fin de l'étape de crible, on a une (énorme) matrice à inverser. Cette matrice a deux parties.
- une sous-matrice de grande dimmension mais creuse avec des entrées minuscules (principalement ± 1),
- quelques colonnes denses dont les éléments sont de taille p, les maps de Schirokauer.

Pour mener à terme l'inversion linéaire, on combine classiquement deux phases :
- une élimination gaussienne structurée (introduite par Lamachia et Odlyzko en 1991) pour réduire significativement la dimension du système à résoudre,
- une résolution itérative comme décrit par Lanczos adaptée à la présence de maps.

- L'élimination gaussienne structurée. L'élimination gaussienne structurée telle que décrite dans [95] permet de réduire la dimension des systèmes à résoudre d'un facteur 1/3. Nos améliorations permettent de réduire cette dimension plus encore. Pour cela, nous essayons de minimiser l'augmentation de poids pendant chacune des éliminations. Lors d'un pivotage dans une ligne de poids ℓ et une colonne de poids plus petit que c, la variation estimée sur le poids total est égale à

$$(c-1)((\ell-1)-1)-\ell=(\ell-2)(c-2)-2.$$

On choisit donc l'entrée avec $(\ell-2)(c-2)$ minimal comme pivot. Après chaque étape, $(l-2)(c-2)$ doit être recalculé mais grâce à une représentation adaptée de la matrice nous pouvons le faire incrémentalement, donc efficacement.

Précisément, nous stockons lignes et colonnes de la matrice pour obtenir rapidement ℓ et c. Pour chaque variable (excepté celles qui apparaissent souvent), nous gardons un pointeur sur la ligne lourde qui la contient et la valeur qui correspond à $(\ell-2)(c-2)$, et nous emmagasinons ces données dans un arbre binaire équilibré.

- L'algorithme de Lanczos. Multiplier la matrice et sa transposée conduit à une matrice symétrique à partir de laquelle on peut définir un produit scalaire. L'algorithme de Lanczos [95] peut être vu comme un procédé d'orthogonalisation pour ce produit scalaire. Il permet ainsi de trouver le vecteur X solution de l'équation matricielle $AX=Y$. Son coût principal est deux fois la dimension de matrice par le coût de la multiplication de la matrice avec un vecteur dont les éléments sont des nombres entiers de taille p. Ajouter quelque colonnes denses en raison des "maps" de Schirokauer ne modifie que peu l'algorithme correspondant.

Le problème majeur de cet algorithme est qu'il se distribue mal, en particulier sur un réseau de stations de travail. Jusqu'à présent, les essais de parallélisation de cet algorithme ont porté pour l'essentiel sur le produit matrice-vecteur. Du coup, ils nécessitent l'utilisation de machines massivement distribuées.

2.2.5 Le calcul d'une instance particulière du logarithme discret

Avant ces travaux, le calcul du logarithme discret d'éléments quelconques de $\mathbb{F}_p$, une fois l'inversion linéaire précédente réalisée, posait problème. La solution théorique consiste à mettre les éléments dont on souhaite le logarithme dans la base de lissité. C'est en pratique bien évidemment non satisfaisant. L'originalité de notre approche est que nous calculons des instances particulières du logarithme discret indépendamment de la phase de crible et d'inversion linéaire.

L'idée naturelle est d'exprimer le logarithme recherché en fonction du logarithme de petits nombres premiers. Simplement, on ne peut utiliser que des nombres premiers qui se décomposent complètement dans les corps de nombres $\mathbb{Q}(\alpha)$ ou $\mathbb{Q}(\beta)$. Dans une situation typique, ces corps de nombres sont de degrés deux et trois. Dans un corps quadratique, environ la moitié des nombres premiers se décomposent en idéaux premiers de degré un, et en prenant en compte celui de degré trois, on augmente assez peu les nombres premiers finalement utilisables. Dans cette voie, rappelons qu'une technique classique pour calculer le logarithme discret d'un élément y particulier est de rechercher deux nombres entiers A et B lisses de taille $\sqrt{p}$ tels que

$$y=\frac{A}{B} \bmod p.$$

On essaie alors de nombreux tels couples (A,B) jusqu'à ce que l'un d'eux soit lisse. En fait, on améliore sensiblement l'efficacité de la recherche en utilisant des techniques de crible. Une fois réduit le réseau $\begin{pmatrix} y & p \\ 1 & 0 \end{pmatrix}$, nous avons alors des nombres entiers A_1, B_1, A_2 et B_2 tels que

$$\forall(\gamma,\delta)\in\mathbb{Z}^2,\ y=\frac{A_1}{B_1}=\frac{A_2}{B_2}=\frac{\gamma A_1+\delta A_2}{\gamma B_1+\delta B_2} \bmod p,$$

et la méthode du crible par vecteurs s'applique pour trouver des nombres entiers γ et δ tels que $\gamma A_1+\delta A_2$ et $\gamma B_1+\delta B_2$ soient simultanément lisses sur l'ensemble des nombres premiers qui se décomposent dans $\mathbb{Q}(\alpha)$ ou $\mathbb{Q}(\beta)$.

L'amélioration proposée permet de se débarrasser de la contrainte sur les nombres premiers utilisés dans le crible. Elle présuppose que le degré du corps de nombres de degré le plus élevé soit un nombre premier et que son groupe de Galois soit cyclique. Pour le cas du degré trois, par exemple, il suffit que le discriminant du polynôme soit un carré. Tout nombre premier non inerte se décompose alors en idéaux premiers de degré 1. Dans ce cas, étant donné un élément y dont on cherche le logarithme discret, on peut essayer de l'écrire sous la forme

$$z = \frac{a_0 + a_1\mu + \cdots + a_d\mu^d}{b_0 + b_1\mu + \cdots + b_d\mu^d} \bmod p,$$

où, en utilisant une réduction de réseau, $a_0, a_1, \ldots, a_d$ et $b_0, b_1, \ldots, b_d$ sont des entiers de taille $O(p^{1/(2d+2)})$ premiers entre eux. Dans ce cas, il n'est pas difficile de montrer que les idéaux principaux $(a_0 + a_1\beta + \cdots + a_d\beta^d)$ et $(b_0 + b_1\beta + \cdots + b_d\beta^d)$ se décomposent en idéaux de degré un dans $\mathbb{Q}(\beta)$. Le bénéfice d'un crible alors réalisé dans $\mathbb{Q}(\beta)$ est que les idéaux premiers facteurs de ces entiers algébriques sont systématiquement dans la base de de lissité de $\mathbb{Q}(\beta)$.

D'un point de vue plus algorithmique, notons que le crible que l'on utilise autorise la présence d'idéaux, certes de taille raisonnable, mais cependant hors de la base de lissité de $\mathbb{Q}(\beta)$. Ces derniers servent alors de "spécial-q" dans un crible annexe faisant en particulier intervenir $\mathbb{Q}(\alpha)$. On peut aussi autoriser des idéaux de grande taille du côté $\mathbb{Q}(\alpha)$, on a ainsi au final un véritable arbre de cribles successifs.

2.3 Quelques résultats

Pour illustrer notre propos, nous décrivons brièvement deux calculs. Un premier calcul pour un nombre premier de 90 chiffres qui a été réalisé en 1998 avec la méthode des entiers de Gauss. Un second calcul réalisé en 2001 pour un nombre premier de 120 chiffres qui utilise le crible algébrique. Ce dernier calcul est toujours le plus gros calcul publié sur le sujet.

2.3.1 90 chiffres par la méthode des entiers de Gauss

Soit

$$p = \lfloor 10^{89}\pi \rfloor + 156137 = 314159265358979323846264338327950288419716939937510582097494459230781640628620899862959619,$$

nous cherchons alors à calculer le logarithme discret en base 2 de

$$y = \lfloor 10^{89}e \rfloor = 271828182845904523536028747135266249775724709369995957496696762772407663035354759457138217.$$

Nous choisissons alors comme polynômes de définition de $\mathbb{Q}(\alpha)$ et $\mathbb{Q}(\beta)$,

$$\begin{aligned} f_\beta(X) &= X^2 + 2, \\ f_\alpha(X) &= -2487489975172435043309669560589929434136978 27X + \\ &\quad 436356663553236108573801834571320046705446681. \end{aligned}$$

Notons que les unités de $\mathbb{Q}(\beta)$ sont $\{\pm 1\}$ et que $h_\beta = 1$.

Avec pour bornes de lissité $B_\alpha = 1299709$ (100000 nombres premiers), et $B_\beta = 350377$ (30000 nombres premiers), un crible de quatre mois sur un PC 180 MHz a conduit à 6663097 équations. Supprimer les relations inutiles conduit finalement à une matrice 976062×674564.

L'élimination Gaussienne structurée ramène alors en un jour ce système à une matrice 61136×61036 et 5683954 entrées. Restait à inverser le système, ce qui fut fait par l'algorithme de Lanczos en trois semaines. On a obtenu ainsi,

$$\begin{aligned} \log_2(3) &= 19748300961167147793248409109777288916563744188 \\ &\quad 34055331610636983877116544539590324530198 44, \\ \log_2(5) &= 14911991564240187116623990082296104898497606799 \\ &\quad 35694758495470585356252119339991916749097892, \\ &\vdots \end{aligned}$$

Pour finir, après neuf heures de calcul, on a trouvé que

$$1299709^{92}y = A/B \bmod p,$$

avec

$$\begin{aligned} A &= 7\cdot 167\cdot 347\cdot 421\cdot 1049\cdot 3067\cdot 3079\cdot 3319\cdot 174077\cdot 293263\cdot 38174761\cdot 82044163, \\ B &= 5^2\cdot 17\cdot 1187\cdot 1877\cdot 2039\cdot 3019\cdot 110863\cdot 207589\cdot 2903639\cdot 5397737\cdot 1064068253. \end{aligned}$$

Et par voie de conséquence, on a

$$\begin{aligned} \log_2(y) = {} & 176713807211421696273204823407162027230205795 \\ & 244991415749384471667791865853837418810109 3. \end{aligned}$$

2.3.2 120 chiffres par le crible algébrique

Soit

$$\begin{aligned} p = \lfloor 10^{119}\pi \rfloor + 207819 = {} & 314159265358979323846264338327950288419716939937510582097494 \\ & 459230781640628620899862803482534211706798214808651328438483, \end{aligned}$$

nous cherchons alors à calculer le logarithme discret en base 2 de

$$\begin{aligned} y = \lfloor 10^{119}e \rfloor = {} & 271828182845904523536028747135266249775724709369995957496696 \\ & 762772407663035354759457138217852516642742746639193200305992. \end{aligned}$$

La phase de crible a consisté à trouver des couples $(a.b)$ tels que l'idéal principal $(a + b\beta)$ soit de norme lisse dans le corps de nombres défini par

$$\beta^3 - 9\beta^2 - 9\beta + 9$$

et tels que $(ac + b\alpha)$ soit de norme lisse dans le corps de nombres défini par

$$\alpha^2 - 737438916792271127953863346119930803308 7\alpha - 333238556260219119547406855509826713348c,$$

où

$$c = 120163929118842727112201927229597987212 5.$$

Ici, le groupe de Galois du polynôme de degré 3 est d'ordre 3. Le corps de nombres correspondant a deux unités fondamentales, $1/12\beta^2 - \beta + 1/4$ et $-1/12\beta^2 + 1/2\beta + 5/4$, son groupe de classe est d'ordre un, son index égal à $2^6 3^2$.

Le crible donna, près quarante jours de calculs sur une machine quadri-processeurs DEC alpha cadencée à 525 MHz, 2685597 équations avec 1242551 inconnues. La base de lissité correspondante était constituée d'idéaux de normes plus petites que 15485870 (un million de nombres premiers) pour le corps de degré 2 ou plus petites que 3497870 (250000 nombres premiers) pour l'autre corps. Puisque les idéaux premiers sont tous principaux pour le corps de degré trois, on a tenu compte explicitement de ces unités pour normaliser les équations. Pour le corps de degré deux, par contre, il nous a fallu normaliser avec deux "maps" de Schirokauer.

L'application d'une élimination Gaussienne structurée pour réduire notre système à 271654 équations en 271552 inconnues avec 22690782 entrées non-nulles fut fait en un jour. La phase critique était l'inversion finale via l'algorithme de Lanczos. Notre version parallélisée de cet algorithme a pris 30 jours pour 4 processeurs. A la fin, nous avions le logarithme pour les générateurs des idéaux de petites normes. En particulier, nous avions des logarithmes pour certains petits nombres premiers.

Par exemple,

$$\begin{aligned} 3 = {} & 2\hat{}288125880933147765096990102563322712059112192935336069483 09 \\ & 244629961378102894412283315317373685769257402738003506902138, \\ 5 = {} & 2\hat{}217557183058115838294593407867074889315520803006202880494 91 \\ & 607941884261289872724504624762374681400333526608185411664140 1, \\ & \vdots \end{aligned}$$

Pour finalement calculer des instances particulières du logarithme discret, on tire avantage du groupe de Galois de $\beta^3 - 12\beta^2 - 9\beta + 12$. Précisément, nous avons trouvé en moins de six heures, utilisant une implémentation naïve, deux entiers algébriques

$$n = 1369196284715334534 65t^2 - 109185518772042207040t - 218010383119442982304$$

et

$$d = 907521772478612632 94t^2 + 59765023811388617 85t + 161979899979266169279,$$

tels que,

$$19^2 y = n/d \bmod p,$$

et tels que, en notation GP-PARI, l'idéal principal (n) est égal à

$$\begin{aligned}
&[[53, [7, 2, 0]\ , 1, 1, [-18, 19, 12]\ ]1] * [[431, [-20, 2, 0]\ , 1, 1, [168, 20, 12]\ ]1] * \\
&[[2179, [140, 2, 0]\ , 1, 1, [-502, -300, 12]\ ]1] * [[16831, [-3928, 2, 0]\ , 1, 1, [-1478, 7836, 12]\ ]1] * \\
&[[156781, [-45467, 2, 0]\ , 1, 1, [15351, -65867, 12]\ ]1] * \\
&[[7691507, [-3118090, 2, 0]\ , 1, 1, [-1592322, -1455347, 12]\ ]1] * \\
&[[5847120361, [-944554227, 2, 0]\ , 1, 1, [-2613536996, 1889108434, 12]\ ]1] * \\
&[[7689099923, [-1979641955, 2, 0]\ , 1, 1, [-1763503409, -3729816033, 12]\ ]1] * \\
&[[14023824873312563677, [2910448673841826685, 2, 0]\ , 1, 1, \\
&\qquad [-4872413772263364932, -5820897347683653390, 12]\ ]1].
\end{aligned}$$

et l'idéal principal (d) est égal à

$$\begin{aligned}
&[[2, [2, 0, 0]\ , 1, 3, [1, 0, 0]\ ]1] * [[3, [1, -1, 1]\ , 3, 1, [2, 2, 0]\ ]3] * \\
&[[19, [8, 2, 0]\ , 1, 1, [-3, 2, -7]\ ]1] * [[1873, [-237, 2, 0]\ , 1, 1, [889, 454, 12]\ ]1] * \\
&[[110359, [36889, 2, 0]\ , 1, 1, [-37268, 36561, 12]\ ]1] * \\
&[[2672473789, [-932319595, 2, 0]\ , 1, 1, [1125968488, -807834619, 12]\ ]1] * \\
&[[626844366559, [-255501920253, 2, 0]\ , 1, 1, [-211702090482, -115840526073, 12]\ ]1] * \\
&[[685495972547, [-197652881054, 2, 0]\ , 1, 1, [-276404069769, -290190210459, 12]\ ]1] * \\
&[[641614040507139551, [33835176915624305, 2, 0]\ , 1, 1, \\
&\qquad [159262188369356649, -67670353831248630, 12]\ ]1].
\end{aligned}$$

Alors, en utilisant une suite de cribles avec des special-q de tailles décroissantes, en particulier en calculant le logarithme discret des générateurs dans le corps de nombres de degré trois des idéaux de normes 7691507, 5847120361, 7689099923, 14023824873312563677 et 25465743776843, 160516256694037129 pour n (et de façon similaire pour d) conduit, grâce à une heure de calcul pour chaque, à

$$\begin{aligned}
y \quad = \quad & 2\hat{}26211228068581138763600862203819182737039076852065697424303 5 \\
& 38038219347876743601868144980494084037374164145286473076508 2.
\end{aligned}$$

En conclusion, le temps dont nous avons eu besoin pour calculer le logarithme discret d'un élément arbitraire modulo un nombre premier à 120 chiffres sur un serveur DEC alpha quadri-processeurs cadencé à 525 MHz est d'approximativement 12 heures pour chaque, une fois le crible (42 jours) et l'algèbre linéaire (30 jours) exécutés.

3 Logarithmes discrets dans $\mathbb{F}_{p^n}$, p petit

Outre la méthode index-calculus générale que nous décrivons au paragraphe 1, on peut distinguer trois familles d'algorithmes pour calculer des logarithmes discrets dans $\mathbb{F}_{2^n}$.

– La méthode de Copersmith (1984) de complexité $L_{p^n}[\frac{1}{3}, c]$ avec $(\frac{32}{9})^{\frac{1}{3}} \leqslant c \leqslant 4^{\frac{1}{3}}$.
– Le crible dans les corps de fonctions dus à Adleman (1994) de complexité $L_{p^n}[\frac{1}{3}, (\frac{64}{9})^{\frac{1}{3}}]$.
– Le crible dans les corps de fonctions amélioré (FFS) de complexité $L_{p^n}[\frac{1}{3}, (\frac{32}{9})^{\frac{1}{3}}]$ avec une première proposition par Adleman et Huang (1999) et une alternative que nous décrivons ici par Joux et Lercier.

Les calculs majeurs réalisés avec ces méthodes sont donnés sur le tableau 3.

Corps	Méthode	Quand	Qui	Commentaires
$\mathbb{F}_{2^{401}}$	Coppersmith	1992	Gordon et McCurley	5 jours sur une nCUBE-2, 1024 processeurs
$\mathbb{F}_{2^{521}}$	FFS	2002	Joux et Lercier	1 mois sur 4 processeurs alpha 525 MHz
$\mathbb{F}_{2^{607}}$	Coppersmith	2002	Thomé	100 PC 450 MHz pendant un an

TABLE 3 – Calculs de logarithmes discrets dans $\mathbb{F}_{2^n}$

3.1 Le crible dans les corps de fonctions

L'idée principale derrière le crible dans les corps de fonctions ("Function Field Sieve") est de "traduire" la méthode du crible algébrique du langage des corps de nombres dans le langage des corps de fonctions algébriques. Il est en effet bien connu que l'on a le dictionnaire

$$\begin{array}{rcl} \text{nombres algébriques} & \to & \text{fonctions algébriques,} \\ \text{idéaux} & \to & \text{places,} \\ \text{groupe des classes} & \to & \text{groupe de Picard,} \\ & \vdots & \end{array}$$

On avait ainsi pour le crible algébrique deux corps de nombres $\mathbb{Q}(\alpha)$ et $\mathbb{Q}(\beta)$, on a maintenant deux corps de fonctions algébriques. Tout d'abord $\mathbb{F}_p(t)$ défini par $\mathbb{F}_p(t)[X]/(\mu_2(t)X - \mu_1(t))$, et ensuite le corps défini par la courbe plane $\mathcal{C}$ de jacobienne $\mathcal{J}$ définie par un polynôme $H(t, X)$ sur $\mathbb{F}_p$ tel que $H(t, \mu(t)) = 0$ dans $\mathbb{F}_{p^n} \simeq \mathbb{F}_p[t]/f(t)$, où $\mu(t) = \mu_1(t)/\mu_2(t)$.

Dans $\mathbb{F}_p(t)$, il est immédiat de passer des places aux fonctions génératrices de ces derniers car elles sont principales et l'on peut réduire simplement dans le corps fini par $(a + bX) \to a + b\mu$. Dans $\mathcal{C}$, par contre, le passage des places aux fonctions ne peut se faire qu'une fois élevé à la puissance le cardinal de la jacobienne de la courbe. On a ainsi pour toute place $\mathfrak{p}$ de $\mathcal{C}$, $\exists \pi_{\mathfrak{p}} \in \mathbb{F}_p(t, X)$, $\mathfrak{p}^h = \mathrm{div}(\pi_{\mathfrak{p}})$ où $h = \#\mathcal{J}(\mathbb{F}_{p^n})$. À une place $\mathfrak{p}$ correspond donc l'élément $\pi_{\mathfrak{p}}(t, \mu)^{1/h}$ de $\mathbb{F}_{p^n}$. La base de lissité se déduit alors de

$$\mathcal{S} = \{\pi \in \mathbb{F}_p[t], \pi \text{ irréductible de degré } < B_\alpha\} \cup \left\{\mathfrak{p}, \begin{array}{l}\text{places irréductibles de degré relatif 1} \\ \text{au dessus de leur restriction à } \mathbb{F}_2(t)\end{array}, \deg(\mathfrak{p}) < B_\beta\right\} \cup \{\text{places à l'infini}\}.$$

L'étape 1 de la méthode générique est alors simplement reformulée comme suit.

Étape 1 : on recherche des couples $(a(t), b(t)) \in \mathbb{F}_p[t]^2$ tels que $a + b\mu$ et $\mathrm{div}(a + bX)$ soient tous deux lisses. Alors

$$\left(v \prod_{\pi \in S} \pi^{e_\pi}\right)^{p-1} = (a + b\mu)^{p-1} = \left(\prod_{\mathfrak{p} \in S} \pi_{\mathfrak{p}}(t, \mu)^{e_{\mathfrak{p}}/h}\right)^{p-1}.$$

3.1.1 Détermination des polynômes

Le coefficient c de la complexité heuristique $L_{p^n}[1/3, c]$ de ces algorithmes est surtout fonction de la taille des coefficients et des degrés des polynômes qui définissent les corps de fonctions utilisés. On dispose de quatre constructions.

- Coppersmith (1984). Soit $\mathbb{F}_{2^n} \simeq \mathbb{F}_2[t]/(f(t))$ où $f(t) = t^n + \lambda(t)$ et $\lambda(t)$ a un degré bas. Alors, on définit avec d est une puissance de deux et $e > n/d$,

$$H(t, X) = X^d + t^{de-n}\lambda(t) \ (\equiv \ X^d + t^{de} \bmod f(t)).$$

Notons que cette construction n'est pas optimale quand n n'est pas une puissance de deux. Cependant la méthode est séduisante en pratique car la moitié des relations nécessaires s'obtient automatiquement (ces dernières proviennent de la factorisation systématique, modulo tout polynôme irréductible de $\mathbb{F}_2$, de $X^d + t^{de-n}\lambda(t)$ en un polynôme de degré un élevé à la puissance d).

Elle conduit à $\sqrt[3]{\frac{32}{9}} \leqslant c \leqslant \sqrt[3]{4}$.

- Adleman (1994). Soit $\mathbb{F}_{p^n} \simeq \mathbb{F}_p[f]/(f(t))$, la méthode comme décrit par Adleman est une adaptation de la méthode de la base μ utilisée pour factoriser des entiers avec GNFS. On choisit un polynôme $\mu(t)$ au hasard de degré $n/(d+1)$ pour un degré d fixé par avance.

On écrit $f(t)$ comme $f(t) = \sum_{i=0}^{d} h_i \mu^i$ et au final les corps de fonctions sont définis par $X - \mu(t)$ et $H(t, X) = \sum_{i=0}^{d} h_i X^i$. Adleman ajoute huit conditions techniques sur H, la plupart d'entre elles pour mieux contrôler les places à l'infini de la courbe définie par $\mathcal{C}$.

Cela donne, $c = \sqrt[3]{\frac{64}{9}}$.

- Adleman et Huang (1999). Soit $\mathbb{F}_{p^n} \simeq \mathbb{F}_p[f]/(f(t))$, Adleman et Huang proposent une construction similaire à celle de SNFS pour factoriser des entiers de forme spéciale.

Soit $f(t) = t^n + \lambda(t)$ où $\lambda(t)$ est de petit degré, alors les polynômes de définition des corps de fonctions algébriques sont simplement $X - t^e$ et

$$H(t, X) \quad = \quad X^d - t^{ed-n}\lambda(t),$$

où $e = \lceil n/d \rceil$ pour un paramètre d fixé par avance. Cela donne,

$$c = \sqrt[3]{\frac{32}{9}}.$$

- Joux et Lercier (2002). L'idée derrière cette dernière construction est d'effectuer le calcul à l'envers.

On fixe d'abord un polynôme $H(t, X)$ de degré d en X et de bas degré en t. On choisit au hasard $\mu_1(t)$ et $\mu_2(t)$ de degrés au plus $\lfloor n/d \rfloor$ jusqu'à ce que $f(t) = \mu_2(t)^d H(t, \mu_1(t)/\mu_2(t))$ soit irréductible de degré n. Le polynôme définissant la droite est alors $\mu_2(t)X - \mu_1(t)$ et $\mathbb{F}_{p^n}$ est alors défini par $\mathbb{F}_p[t]/(f(t))$.

On combine cette construction avec l'utilisation de courbes $C_{a,b}$ comme suggéré par Matsumoto (1999). Dans ce cas, on a une unique valuation à l'infini. On a aussi le résultant en X de $H(t, X)$ et $\mu_2(t)X - \mu_1(t)$ qui est exactement égal à $f(t)$. Cela donne,

$$c = \sqrt[3]{\frac{32}{9}}.$$

3.1.2 Le crible par vecteurs

Les expérimentations que nous avons menées se sont restreintes à $\mathbb{F}_{2^n}$. Les techniques de crible sont alors assez similaires à celle utilisées pour $\mathbb{F}_p$, excepté que les entiers manipulés par les ordinateurs représentent des polynômes irréductibles de $\mathbb{F}_2[t]$ et plutôt que de parcourir les cases du tableau de crible de façon croissante, on utilise un code de Gray.

Ainsi, pour chaque polynôme irréductible $q(t)$ (ou de façon similaire pour les places irréductibles de degré relatif un de $\mathcal{C}$ considérées), on construit le sous-réseau des couples $(a(t), b(t))$ tels que $q(t)$ divise $b(t)^d H(t, a(t)/b(t))$. À chaque $q(t)$ correspond un réseau distinct. Tout comme $\mathbb{F}_p$, cependant, il est important d'utiliser une base réduite du réseau pour le parcourir efficacement. Cependant, les algorithmes de réduction de réseau sur $\mathbb{Q}$ ne s'adaptant pas comme tel, on propose donc ci-dessous un algorithme de réduction ad-hoc pour des réseaux de dimension 2 sur $\mathbb{F}_2[t]$.

Algorithme `GF2Gauss`
Réduction d'un réseau de dimension 2 sur $\mathbb{F}_2[t]$.
INPUT: Une base du réseau $(\vec{u}, \vec{v})$ avec $\vec{u} = (u_1, u_2)$ et $\vec{v} = (v_1, v_2)$.
OUTPUT: Une base réduite.

1. do {
2. Let $\delta_1 = \deg v_1 - \deg u_1$, $\delta_2 = \deg v_2 - \deg u_2$.
3. Let $\vec{w}^{(1)} = \vec{v} - t_1^{\delta} \cdot \vec{u}$, $\vec{w}^{(2)} = \vec{v} - t_2^{\delta} \cdot \vec{u}$.
4. **if** $\max(\deg w_1^{(1)}, \deg w_2^{(1)}) < \max(\deg w_1^{(2)}, \deg w_2^{(2)})$ **then** let $\vec{w} = \vec{w}^{(1)}$ **else** let $\vec{w} = \vec{w}^{(2)}$.
5. **if** $\max(\deg v_1, \deg v_2) > \max(\deg w_1, \deg w_2)$ **then** let $\vec{v} = \vec{w}$ and declare the loop as active.
6. **if** $\max(\deg u_1, \deg u_2) \geq \max(\deg v_1, \deg v_2)$ **then** exchange $\vec{u}$ and $\vec{v}$.
7. } **while** the loop is not declared active for two consecutive executions.
8. return $(\vec{u}, \vec{v})$

3.1.3 L'algèbre linéaire

L'algèbre linéaire est menée de façon identique au cas $\mathbb{F}_p$ (cf. paragraphe 2.2.4).

3.1.4 Le calcul d'un logarithme arbitraire

C'est ici plus simple que pour $\mathbb{F}_p$ puisque les unités sont très facilement calculables dans un corps de fonctions (il s'agit pour l'essentiel des éléments de $\mathbb{F}_p^*$ et des places à l'infini). L'idée principale est d'écrire $z(t)$ comme $z_1(t)/z_2(t)$ avec $z_1(t)$ et $z_2(t)$ lisses. On réduit donc le réseau $\left(\begin{smallmatrix} z & f \\ 1 & 0 \end{smallmatrix}\right)$, ce qui conduit à une base de la forme $\left(\begin{smallmatrix} A_1 & A_2 \\ B_1 & B_2 \end{smallmatrix}\right)$ et on crible jusqu'à trouver un couple (α, β) tel que $z_1 = \alpha A_1 + \beta A_2$ et $z_2 = \alpha B_1 + \beta B_2$ soient $L_{p^n}[\frac{2}{3}]$-lisses.

Éventuellement, on peut calculer le logarithme de grands facteurs, ceux de taille $L_{p^n}[\frac{2}{3}]$ en les utilisant comme special-q dans un second crible afin de rechercher des polynômes $L_{p^n}[\frac{1}{3}]$-lisses et ainsi tirer partie des précalculs faisant intervenir la base de lissité. Cette stratégie, analogue à celle pour le crible algébrique (cf. paragraphe 2.2.5), est étudiée par Schirokauer dans [148].

3.2 Un exemple, $\mathbb{F}_{2^{521}}$

Pour noter agréablement les polynômes sur $\mathbb{F}_2[t]$, nous utilisons la notation $\overline{\nu}$ où ν est un entier hexadécimal. Par exemple, $\overline{\mathtt{b}}$ représente $t^3 + t + 1$. Posons ainsi

$$H(t, X) = X^5 + X + t + 1,$$

et

$$\mu_2(t)X + \mu_1(t) = \overline{\mathtt{1b92c17dec4c4cf4f5ab9c1e86f}}X + \overline{\mathtt{d0e134790925d9e08}}.$$

Leur résultant, $f(t)$, est un polynôme irréductible de degré 521 qui nous permet de représenter $\mathbb{F}_{2^{521}}$. On choisit alors comme base de lissité pour le crible,

- 300000 places irréductibles de degré relatif un au-dessus de leurs restrictions à $\mathbb{F}_2(t)$ du côté du polynôme de degré 5,
- 210870 places irréductibles du côté du polynôme de degré 1.

Le crible a fourni, après trois semaines de calcul sur une machine quadri-processeurs DEC alpha 525 MHz un total de 472121 équations avec 450940 inconnues. Après l'élimination Gaussienne structurée, il restait 197039 équations en 196939 inconnues comprenant 12220108 entrées non-nulles (62 entrées par équation).

Appliquer l'algorithme de Lanczos à ce système conduit en dix jours aux logarithmes de petits polynômes irréductibles. Typiquement, on trouve,

$$\log_t(t+1) = 946815771521222940761751735986503246062188885220190526391080148799898588434$$
$$5864952201320754968825133615526417923163653891428634582550637955161092146219401 59,$$

ou encore,

$$\log_t(t^2+t+1) = 409945320335775766828444393363213401554375603879607112148806279189823617301$$
$$300239132485640738107949052818943078142206215533143595141990328387727782201876189 1.$$

Pour terminer, on a calculé le logarithme discret de

$$e(t) = \overline{\lfloor 2^{519} e \rfloor} = t^{520} + t^{518} + \ldots + t^6 + t^3.$$

En particulier, on trouve en quelques heures,

$$z_1(t) = \overline{\texttt{17acf35dc9215}} \times \overline{\texttt{33cab5311}} \times \overline{\texttt{83b6db37}} \times \overline{\texttt{88af29f}} \times \overline{\texttt{4c99eb3}} \times$$
$$\overline{\texttt{1a22cdd}} \times \overline{\texttt{debb79}} \times \overline{\texttt{6358f}} \times \overline{\texttt{304f}} \times \overline{\texttt{6b5}} \times \overline{\texttt{41b}} \times \overline{\texttt{75}} \times \overline{2}^4$$

et

$$z_2(t) = \overline{\texttt{41edc78c5127}} \times \overline{\texttt{75a6c0fe253}} \times \overline{\texttt{b66ac13d5}} \times \overline{\texttt{d422507}} \times \overline{\texttt{b0b0e11}} \times$$
$$\overline{\texttt{d2c45}} \times \overline{\texttt{81869}} \times \overline{\texttt{54e1}} \times \overline{\texttt{a85}} \times \overline{\texttt{409}} \times \overline{\texttt{25f}} \times \overline{\texttt{fd}} \times \overline{\texttt{3b}} \times \overline{\texttt{7}} \times \overline{\texttt{3}}$$

tels que

$$\overline{\texttt{3fffcd}}^{43} \times e(t) = z_1(t)/z_2(t) \bmod f(t).$$

Et on a finalement,

$$\log_t e(t) = 263247762193834129884994702428538360289317407093273177190025600958418025325 46$$
$$548170764837586429284565024547468908202520438767346267799208009538061094578743 58.$$

Il est assez aisé d'extrapoler à une année sur un quadri-processeurs 525MHz le temps nécessaire pour un calcul dans $\mathbb{F}_{2^{607}}$ par cette méthode. Notons que ce temps est bien moindre que celui d'un calcul mené pour ce corps avec la méthode de Coppersmith (cf. tableau 3).

Cardinalités de courbes elliptiques

— *L'algorithme SEA*

Comme souligné par Elkies au tout début de [46], "the problem of calculating the trace of an elliptic curve over a finite field has attracted considerable interest in recent years". Grâce à la contribution de beaucoup de gens dans ce domaine, on dispose maintenant d'algorithmes raisonnables pour cela.

En fait, avec la publication en 1985 d'un algorithme déterministe de complexité en $O(\log^{5+\varepsilon} q)$ pour calculer le nombre de points d'une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ [152], Schoof offrait une alternative fort alléchante aux méthodes de complexité en $O(q^{1/4}\log^2 q)$ connues jusqu'alors. En effet, avec par exemple l'algorithme "des pas de bébé et de géant" de Shanks [157], il n'est pas envisageable d'avoir $q > 10^{30}$. Certains pensaient même que ce problème était un problème difficile puisque, finalement, cela ne revient-il pas à calculer le logarithme discret de l'élément neutre ?

L'algorithme de Schoof n'aurait pas pris l'importance que nous lui connaissons aujourd'hui sans d'une part, les travaux d'Atkin qui, au travers de deux communications importantes [5, 6], montre comment calculer le nombre de points d'une courbe définie sur $\mathbb{F}_{10^{275}+693}$, et d'autre part, les travaux d'Elkies [45] qui complètent ceux d'Atkin. À l'invitation d'Atkin qui écrit en 1992, "since there is now considerable variety in the mathematics and programming involved in the problem, it is to be hoped that more people will try their hand at it", de nombreuses améliorations ont suivi. Tous ces travaux forment le corps d'un algorithme probabiliste appelé l'algorithme `SEA` (Schoof, Elkies, Atkin) dont la complexité est de $O(\log^{4+\varepsilon} q)$.

Notre but est ici de donner un panorama relativement complet de ces idées.

1 Module de Tate

1.1 Courbes sur un corps algébriquement clos

Dans ce paragraphe qui reprend le premier chapitre de [65], k est un corps algébriquement clos.

1.1.1 Variétés affines

Soit $\mathbb{A}_k^n$ l'espace affine de dimension n sur k, i.e l'ensemble des n-uplets à coordonnées dans k. À chaque idéal I de $A = k[x_1, \ldots, x_n]$, on associe un *ensemble algébrique affine*, $X_I = \{P \in \mathbb{A}_k^n \mid f(P) = 0\ \forall f \in I\}$. Les ensembles algébriques affines sont les fermés de la topologie dite de Zariski.

À l'inverse, l'idéal $I(X)$ associé à un ensemble algébrique X est l'idéal des polynômes qui s'annulent sur X. Un ensemble algébrique X est appelé une *variété affine* si $I(X)$ est un idéal premier. On appelle *variété quasi-affine* un ouvert d'une variété affine.

On appelle anneau des fonctions régulières de X et on note $A(X)$ le quotient $A/I(X)$. C'est un anneau intègre dont le corps des fractions noté $k(X)$ est appelé *corps de fonctions* de X. La *dimension* de X est le degré de transcendance de $k(X)$. À tout point P de X, on associe l'idéal $\mathfrak{m}_P \subset A(X)$ des fonctions de $A(X)$ qui s'annulent en P et on note $\mathcal{O}_P$ le localisé $A(X)_{\mathfrak{m}_P}$, l'anneau des fonctions régulières en P. Il y a bijection entre les idéaux maximaux $\mathfrak{m}_P$ de $A(X)$ et les points P de X.

Si X est une variété quasi-affine de dimension d et $P \in X$ un point, on dit que X est lisse en P si l'idéal $I(X)$ est engendré par des fonctions $f_1, \ldots, f_N$ telles que $f_i(P) = 0$ et si la famille des formes linéaires différentielles $(D_P(f_i))_{1 \leqslant i \leqslant N}$ a pour rang $n - d$, la codimension de X.

1.1.2 Variétés projectives

On note $\mathbb{P}_k^n$ l'espace projectif de dimension n, c'est-à-dire le quotient de $\mathbb{A}_k^{n+1}$ par la relation d'équivalence induite par la multiplication par des unités. On ainsi la décomposition $\mathbb{P}_k^n = \mathbb{A}_k^n \cup \mathbb{P}_k^{n-1}$ où $\mathbb{A}_k^n$ et $\mathbb{P}_k^{n-1}$ sont obtenus en fixant la i-ième composante des éléments respectivement à 1 et 0. On considère alors l'anneau $S = k[x_0, \ldots, x_n]$ des polynômes en $n+1$ indéterminées gradué par le degré total. Pour des idéaux de S engendrés par des polynômes homogènes, on peut définir, comme au-dessus, des *ensembles algébriques projectifs* et lorsque ces derniers sont premiers, des *variétés projectives* V. L'intersection $V \cap \mathbb{A}_k^n$ pour un choix de $\mathbb{A}_k^n$ est appelé la partie affine de V et $V \cap \mathbb{P}_k^n$ est appelé l'ensemble des points à l'infini. Le corps de fonctions de V est le corps de fonctions de la variété affine $V \cap \mathbb{A}_k^n$. On associe alors à tout point P l'anneau $\mathcal{O}_P \subset k(X)$ des fonctions régulières en P.

On peut définir l'anneau de coordonnées S/I qui est l'algèbre graduée des formes homogènes sur la variété projective X. Un ouvert d'une variété projective est appelé une variété quasi-projective. Dans ce texte, une *variété algébrique* est une variété quasi-affine ou quasi-projective. Un *morphisme* de variétés algébriques $\phi : X \longrightarrow Y$ est une application continue pour la topologie de Zariski qui induit en tout point $P \in X$ tel que $\phi(P) = Q \in Y$ une application de $\mathcal{O}_Q$ dans $\mathcal{O}_P$. Si la variété Y est affine, un morphisme se réduit à un morphisme d'algèbre de $\mathcal{O}(Y)$ dans $\mathcal{O}(X)$. Une *application rationnelle* de X vers Y est un morphisme défini seulement d'un ouvert non-vide de X vers un ouvert non-vide de Y. Une application rationnelle est dite dominante si son image contient un ouvert non-trivial de Y.

1.1.3 Courbes algébriques

Une *courbe affine (resp. projective)* est une variété affine (resp. projective) de dimension 1. Une courbe est lisse si elle est lisse en tout point.

L'anneau local d'une courbe C en un point lisse P est un anneau de valuation discrète. Son unique idéal maximal $\mathfrak{m}_P$ est l'ensemble des fonctions qui s'annulent en P. On peut alors définir la valuation ord_P par $\mathrm{ord}_P : \mathcal{O}_P \longrightarrow \mathbb{N} \cup \{\infty\}, f \longrightarrow \mathrm{ord}_P(f) = \max_{d \in \mathbb{Z}}\{f \in \mathfrak{m}_P^d\}$ et ord_P peut être facilement étendu à $k(C)$ en définissant $\mathrm{ord}_P(f/g) = \mathrm{ord}_P(f) - \mathrm{ord}_P(g)$. Une uniformisante pour C en P est une fonction $t \in k(C)$ avec $\mathrm{ord}_P(t) = 1$.

Étant donné un corps de fonctions, il existe toujours une courbe projective lisse C, appelée le modèle non-singulier, de corps de fonctions isomorphe au corps donné. Pour une telle courbe, on peut définir le diviseur d'une fonction f par la somme formelle $\mathrm{div} f = \sum_{P \in C} \mathrm{ord}_P(f)P$. Plus largement, l'ensemble de telles sommes formelles finies est appelé le groupe des diviseurs de X. La somme des valuations d'un diviseur est appelée son degré. L'ensemble des diviseurs de fonctions (appelés diviseurs principaux) est un sous-groupe du groupe des diviseurs. Le quotient de ces deux groupes est le groupe de Picard $\mathrm{Pic}(X)$. Comme le degré des diviseurs principaux est nul, l'application degré est définie sur $\mathrm{Pic}(X)$ et son noyau est le sous-groupe $\mathrm{Pic}^0(X)$.

1.1.4 Variétés abélienes

Un groupe algébrique est une variété algébrique munie d'une structure de groupe compatible avec la structure de variété. Ce qui signifie que les opérations addition ou inverse sont régulières et que l'élément neutre est un point rationnel de la variété.

Soit O un point d'une variété algébrique X. Le résultat fondamental de la théorie des variétés Jacobiennes est qu'il existe un groupe algébrique J et un monomorphisme $\varphi : X \longrightarrow J$ tel que $\varphi(O) = 0$ et que l'application $\Phi : \mathrm{Div}^0(X) \longrightarrow J$ soit surjective et ait pour noyau le groupe des diviseurs principaux. On identifie ainsi $\mathrm{Pic}^0(X)$ avec J.

La construction de la Jacobienne repose sur le théorème de Riemann-Roch. Pour tout diviseur D, on considère l'espace vectoriel des fonctions $\mathcal{L}(D) = \{f \in k(C)^* \mid \mathrm{div}(f) + D \geqslant 0\} \cup \{0\}$. C'est un espace de dimension finie, on note sa dimension $\ell(D)$ et on a $\ell(D) \geqslant \deg(D) - g + 1$.

1.1.5 Isogénies et module de Tate

Un homomorphisme Φ d'une variété abélienne A vers une variété abélienne B est appelé une isogénie s'il est surjectif et si son noyau est fini. Les dimensions de A et B sont alors égales.

L'isogénie la plus simple est la multiplication par un entier n qui à un point de a de A associe la somme de n fois a. Elle est notée $[n]_A$. Le noyau de $[n]_A$ est appelé la n-torsion et est noté $A[n]$. Quand n est égal à ℓ^k, une puissance de nombre premier ℓ distinct de la caractéristique du corps, $A[n]$ est alors isomorphe à $(\mathbb{Z}/n\mathbb{Z})^{2g}$ où g est égale à la dimension de A. Le module de Tate est défini pour un nombre premier ℓ comme la limite projective de la ℓ^k-torsion, c'est-à-dire

$$T_\ell(A) = \lim_{\leftarrow k} A[\ell^k].$$

Puisque $A[\ell^k]$ est isomorphe à $(\mathbb{Z}/\ell^k\mathbb{Z})^{2g}$, $T_\ell(A)$ est isomorphe à $\mathbb{Z}_\ell^{2g}$ où $\mathbb{Z}_\ell$ est l'ensemble des entiers ℓ-adiques.

1.2 Courbes elliptiques sur $\mathbb{C}$

Se placer sur $\mathbb{C}$ est un angle intéressant pour aborder la théorie des courbes elliptiques. Nous rappelons ici quelques résultats qui peuvent être trouvés dans [159].

1.2.1 Tores complexes et fonctions doublement périodiques

Soit Λ, un *réseau* de $\mathbb{C}$, c'est-à-dire un sous-groupe discret de $\mathbb{C}$ non-nul et non-isomorphe à $\mathbb{Z}$. Il peut être défini par $\mathbb{Z}\omega_1 + \mathbb{Z}\omega_2$ avec $\omega_1, \omega_2 \in \mathbb{C}$ et $\omega_2/\omega_1 \notin \mathbb{R}$. Un *parallélogramme fondamental* pour Λ est un ensemble de la forme $D = \{a + t_1\omega_1 + t_2\omega_2, 0 \leqslant t_1, t_2 < 1\}$ où $a \in \mathbb{C}$. Il y a alors bijection entre D et le *tore* défini par le quotient de $(\mathbb{C}, +)$ par Λ.

Une fonction méromorphe f sur $\mathbb{C}$ est une *fonction elliptique* de périodes Λ si pour tout $\omega \in \Lambda$, $f(z + \omega) = f(z)$. En particulier, une fonction elliptique peut-être vue comme une fonction méromorphe du tore et l'ensemble de ces fonctions forme clairement un corps. Observons aussi qu'une fonction elliptique sans pôle ou sans zéro est constante (théorème de Liouville). Aussi, la somme des résidus d'une fonction elliptique sur un parallélogramme fondamental (ne rencontrant pas ses pôles) est nul (théorème des résidus).

À ce stade, il est naturel d'introduire la fonction $z \longrightarrow \sum_{\omega\in\Lambda} 1/(z-\omega)^3$ car cette série converge absolument uniformément dans tout compact disjoint de Λ et définit une fonction elliptique impaire avec un pôle triple en tout point de Λ. Son intégration, à un terme correctif près, qui garantit la convergence uniforme absolue sur tout compact disjoint de Λ, conduit à la fonction elliptique avec pôles doubles en tout point de Λ appelée la fonction $\wp$ de Weierstrass,

$$\wp(z) = \frac{1}{z^2} + \sum_{\omega\in\Lambda-\{0\}} \frac{1}{(z-\omega)^2} - \frac{1}{\omega^2}.$$

En raisonnant sur les zéros et les pôles z_i d'une fonction elliptique paire $f(z)$, il est toujours possible d'exhiber un produit de fonctions $\wp(z) - \wp(z_i)$ dont le quotient avec f soit elliptique et sans pôle, donc constante. Le corps des fonctions elliptiques paires n'est donc autre que $\mathbb{C}(\wp)$ et, plus largement, on peut montrer le corps des fonctions elliptiques est $\mathbb{C}(\wp, \wp')$.

Comme $\wp'^2(z)$ est une fonction paire, il est donc naturel de chercher à l'exprimer en fonction de $\wp(z)$. On trouve ainsi, une fois posé $G_k = \sum_{\omega\in\Lambda-\{0\}} 1/\omega^{2k}$, que

$$\wp'^2 = 4\wp^3 - 60G_2\wp - 140G_3.$$

L'application $\phi : z \to (\wp(z), \wp'(z))$ est donc une bijection entre le tore défini par $\mathbb{C}/\Lambda$ et l'ensemble des points de la courbe plane projective d'équation $Y^2Z = 4X^3 - G_2XZ^2 - 143G_3Z^3$ de courbe affine E associée, $y^2 = 4x^3 - G_2x - G_3$. Ces courbes sont appelées *courbes elliptiques* et l'existence de ϕ (isomorphisme de surfaces de Riemann) montre qu'elles sont non-singulières, en particulier $20G_2^3 - 49G_3^2$ est non-nul. Le corps $\mathbb{C}(\wp, \wp')$ est donc isomorphe au corps des fractions de l'anneau $\mathbb{C}[x, y]/(y^2 - 4x^3 + G_2x + G_3)$ des fonctions régulières de E.

1.2.2 Loi d'addition

À ce stade, on peut exprimer la loi d'addition sur le tore $\mathbb{C}/\Lambda$ en termes géométriques sur la courbe E. Pour cela, définissons rapidement un diviseur D de $\mathbb{C}/\Lambda$ comme une combinaison formelle finie de points de $\mathbb{C}/\Lambda$, à coefficients dans $\mathbb{Z}$. Il est noté $\sum_{P\in\mathbb{C}/\Lambda} e_P[P]$. Le degré d'un diviseur D est la somme des coefficients. L'ensemble des diviseurs de degré 0, noté $\mathrm{Div}^0(\mathbb{C}/\Lambda)$, est un sous-groupe de l'ensemble des diviseurs $\mathrm{Div}(\mathbb{C}/\Lambda)$. À toute fonction méromorphe f de $\mathbb{C}/\Lambda$, on associe naturellement le diviseur $\mathrm{Div}\, f$ dont les points sont les zéros et le pôles de f et les coefficients, les valuations correspondantes. Un tel diviseur est appelé un diviseur principal. On peut alors montrer l'exactitude de la suite d'applications

$$1 \longrightarrow \mathbb{C}^* \longrightarrow \mathbb{C}(\wp,\wp')^* \longrightarrow \mathrm{Div}^0(\mathbb{C}/\Lambda) \longrightarrow \mathbb{C}/\Lambda \longrightarrow 0.$$

L'exactitude en $\mathbb{C}(\wp,\wp')$ résulte du théorème de Liouville. L'exactitude en $\mathrm{Div}^0(\mathbb{C}/\Lambda)$ revient à dire que les diviseurs de degré 0 de somme 0 sont les diviseurs de fonctions elliptiques. On déduit ainsi que le groupe de Picard $\mathrm{Pic}^0(\mathbb{C}/\Lambda)$ de $\mathbb{C}/\Lambda$, défini comme le quotient de $\mathrm{Div}^0(\mathbb{C}/\Lambda)$ par le groupe des diviseurs principaux, est isomorphe à $\mathbb{C}/\Lambda$. Avec O, le point à l'infini de la courbe projective, l'isomorphisme associé est simplement, $P \longrightarrow [P]-[O]$. Expliciter à partir de cet isomorphisme l'addition sur le tore conduit à la loi d'addition bien connue. À deux points P et Q, on associe le symétrique R par rapport à l'axe des abscisses du point intersection S de la droite D passant par P et Q avec la courbe. Lorsque les deux points sont égaux, on prend pour D la tangente en P à la courbe (cf. figure 1 page 3).

1.2.3 Morphisme entre tores complexes

Soient deux courbes elliptiques projectives E_1 et E_2 associées à deux tores $\mathbb{C}/\Lambda_1$ et $\mathbb{C}/\Lambda_2$. Une application holomorphe de $\mathbb{C}/\Lambda_1$ vers $\mathbb{C}/\Lambda_2$ qui envoie 0 vers 0 est un morphisme du groupe E_1 vers E_2 et si elle n'est pas constante, elle est surjective et on l'appelle une *isogénie*. On peut alors montrer que l'ensemble des isogénies de E_1 vers E_2 est un groupe commutatif isomorphe au sous-ensemble des complexes α tels que $\alpha\Lambda_1 \subset \Lambda_2$. Pour le sens direct, il n'est pas difficile de voir que lorsque α vérifie cette dernière condition, l'application $[\alpha] : z \longrightarrow \alpha z$ est une isogénie. À l'inverse, il faut relever sur $\mathbb{C}$ l'isogénie considérée pour se rendre compte que sa dérivée y est holomorphe et elliptique, donc constante.

Ainsi une isogénie Φ non-nulle a pour noyau le sous-groupe fini $\alpha^{-1}\Lambda_2/\Lambda_1$. Le degré de Φ noté $\deg\Phi$ est le cardinal du noyau de Φ. On a $\deg(\Phi) = [\Lambda_2 : \alpha\Lambda_1] = [\alpha^{-1}\Lambda_2 : \Lambda_1]$. On obtient une injection du corps de fonctions $\mathbb{C}(\Lambda_2)$ dans $\mathbb{C}(\Lambda_1)$,

$$\begin{array}{rccc} \Phi^* & : & \mathbb{C}(\Lambda_2) & \longrightarrow & \mathbb{C}(\Lambda_1), \\ & & f & \longrightarrow & f\circ\Phi\ . \end{array}$$

Cette injection définit une extension galoisienne de degré $\deg\Phi$. À l'inverse, on peut montrer que toute injection de $\mathbb{C}(\Lambda_2)$ dans $\mathbb{C}(\Lambda_1)$ définit un morphisme holomorphe de $\mathbb{C}/\Lambda_1$ dans $\mathbb{C}/\Lambda_2$. Par ailleurs, Φ induit un morphisme de $\mathrm{Div}\, E_1$ dans $\mathrm{Div}\, E_2$ qui conserve le degré, $\sum_{P\in\mathbb{C}/\Lambda} e_P[P] \longrightarrow \sum_{P\in\mathbb{C}/\Lambda} e_P[\Phi(P)]$. On peut montrer en utilisant la norme de l'extension $\mathbb{C}(E_1)/\mathbb{C}(E_2)$ que l'image d'un diviseur principal est un diviseur principal. L'isogénie Φ induit ainsi un morphisme des groupes $\mathrm{Pic}^0(E_1)$ dans $\mathrm{Pic}^0(E_2)$, ou encore entre E_1 et E_2.

Soit d le degré de Φ, alors $\Lambda_1 \subset \alpha^{-1}\Lambda_2 \subset d^{-1}\Lambda_1$ et donc $\Lambda_2 \subset d^{-1}\alpha\Lambda_1$. Il existe donc une isogénie

$$\begin{array}{rccc} \widehat{\Phi} & : & \mathbb{C}/\Lambda_2 & \longrightarrow & \mathbb{C}/\Lambda_1, \\ & & z & \longrightarrow & d\alpha^{-1}z\ . \end{array}$$

Alors, $\widehat{\Phi}\circ\Phi$ est égale à la multiplication par d sur $\mathbb{C}/\Lambda_1$, notée $[d]_1$. De même $\Phi\circ\widehat{\Phi} = [d]_2$.

Si on spécialise aux endomorphismes d'une courbe elliptique E, cela signifie que cet ensemble noté $\mathrm{End}(E)$ est un anneau pour les lois $(+,\circ)$ qui est muni d'une involution $\widehat{\ }$. En fait, $\mathrm{End}(E)$ est l'ensemble des complexes α tels que $\alpha\Lambda \subset \Lambda$. Pour un isomorphisme, on a même $\alpha\Lambda = \Lambda$. D'ailleurs, deux réseaux homothétiques conduisent à des courbes isomorphes. On peut donc toujours se ramener au cas $\Lambda = \mathbb{Z}+\tau\mathbb{Z}$ avec $\tau = \omega_2/\omega_1$ et la condition $\alpha\Lambda = \Lambda$ signifie qu'il existe quatre entiers a, b, c et d tels que $\alpha = a+b\tau$, $\alpha\tau = c+d\tau$. On a donc $\mathrm{End}(E)$ qui est isomorphe à $\mathbb{Z}$ ou à un ordre de l'anneau des entiers de $\mathbb{Q}(\tau)$ dans le cas très particulier où τ est solution d'une équation de degré 2 à coefficients entiers. Dans ce dernier cas, on dit que la courbe est à multiplication complexe.

1.3 Courbes elliptiques sur les corps finis

Par soucis d'effectivité, nous nous proposons d'aborder sous un angle résolument calculatoire les quelques résultats issus de [159] dont on a besoin pour comprendre l'algorithme `SEA`.

Définition 2. *Soit $\mathbb{K}$ un corps, l'équation*

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6 \text{ avec } (a_1, a_2, a_3, a_4, a_6) \in \mathbb{K}^5$$

définit une courbe elliptique si $\Delta_E = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$ est non-nul, une fois posé

$$d_2 = a_1^2 + 4a_2,\ d_4 = 2a_4 + a_1a_3,\ d_6 = a_3^2 + 4a_6, d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2.$$

L'ensemble de ses points est

$$E(\mathbb{K}) = \{O_E\} \cup \{(X, Y) \in \mathbb{K}^2, Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6\}.$$

On appelle j-invariant d'une courbe elliptique $E/\mathbb{K}$, $j_E = \frac{c_4^3}{\Delta_E}$ où $c_4 = d_2^2 - 24d_4$. Deux courbes isomorphes ont le même invariant (l'inverse est vrai pour les corps algébriquement clos).

1.3.1 Loi de groupe

Le groupe de Picard associé à une courbe elliptique permet au final d'additionner des points. Une formulation de la loi d'addition correspondante suit.

Théorème 5. *L'ensemble $E(\mathbb{K})$ est un groupe abélien si on le munit de la loi d'addition + suivante :*
- *pour tout point $P = (x_P, y_P)$ de $E(\mathbb{K})$, $P + O_E = O_E + P = P$ et $-P = (x_P, -y_P - a_1x_P - a_3)$.*
- *pour tous points $P = (x_P, y_P)$ et $Q = (x_Q, y_Q)$ de $E(\mathbb{K})$ avec $P \neq -Q$, le point $P + Q$ a pour coordonnées (x_{P+Q}, y_{P+Q}) avec* $\begin{cases} x_{P+Q} &= \lambda^2 + a_1\lambda - a_2 - x_P - x_Q, \\ y_{P+Q} &= -(\lambda + a_1)x_{P+Q} - \nu - a_3, \end{cases}$

$$\text{où } \lambda = \begin{cases} \dfrac{y_Q - y_P}{x_Q - x_P} & \text{si } P \neq Q, \\ \dfrac{3x_P^2 + 2a_2x_P + a_4 - a_1y_P}{2y_P + a_1x_P + a_3} & \text{sinon,} \end{cases} \qquad \text{et } \nu = y_P - \lambda x_P.$$

Un résultat de Hasse-Weil borne le nombre de points de ces groupes lorsque $\mathbb{K}$ est un corps fini.

Théorème 6.

$$\#E(\mathbb{F}_q) = q + 1 - c \text{ with } |c| \leqslant 2\sqrt{q}.$$

1.3.2 Module de Tate

Pour $\ell \in \mathbb{Z}$, on définit, $[\ell]_E : E(\bar{\mathbb{F}}_q) \longrightarrow E(\bar{\mathbb{F}}_q),\ P \mapsto \ell P$. On note $E[\ell]$ son noyau,

$$E[\ell] = \{O_E\} \cup \left\{(X, Y) \in E(\bar{\mathbb{F}}_q), [\ell]_E(X, Y) = O_E\right\},$$

Il existe $\psi_\ell(X, Y) \in \mathbb{K}[X, Y]$ tel que $E[\ell] = \{O_E\} \cup \left\{(X, Y) \in E(\bar{\mathbb{F}}_q), \psi_\ell(X, Y) = 0\right\}$. Le polynôme ψ_ℓ est de degré au plus $(\ell^2 - 1)/2$ en X et 0 en Y si ℓ est impair ou $(\ell^2 - 2)/2$ en X et 1 en Y si ℓ est pair. Les polynômes de division $\psi_\ell(X, Y)$ d'une courbe elliptique E sont donnés par

$$\psi_\ell(X, Y) = \begin{cases} (2Y + a_1X + a_3)f_\ell(X) & \text{si } \ell \text{ est pair,} \\ f_\ell(X) & \text{sinon,} \end{cases}$$

où

$$f_0(X) = 0,\ f_1(X) = 1,\ f_2(X) = 1,$$
$$f_3(X) = 3X^4 + d_2X^3 + 3d_4X^2 + 3d_6X + d_8,$$
$$f_4(X) = 2X^6 + d_2X^5 + 5d_4X^4 + 10d_6X^3 + 10d_8X^2 + n(d_2d_8 - d_4d_6)X + (d_4d_8 - d_6^2),$$

et, en posant $F(X) = 4X^3 + d_2X^2 + 2d_4X + d_6$,
- $f_{2\ell} = f_\ell(f_{\ell+2}f_{\ell-1}^2 - f_{\ell-2}f_{\ell+1}^2)$,
- $f_{2\ell+1} = \begin{cases} F^2f_{\ell+2}f_\ell^3 - f_{\ell-1}f_{\ell+1}^3 & \text{si } \ell \text{ est pair,} \\ f_{\ell+2}f_\ell^3 - f_{\ell-1}f_{\ell+1}^3F^2 & \text{sinon.} \end{cases}$

- Exemple de points de ℓ-torsion. La courbe $E/\mathbb{F}_{97}\ : Y^2+67XY+87Y=X^3+63X^2+3X+36$, est isomorphe via $(X,Y)\mapsto(62X+1,57X+75Y+20)$ à $Y^2=X^3+12X+79$. Alors,

$$f_7(X)=7\,X^{24}+10\,X^{22}+12\,X^{21}+66\,X^{20}+39\,X^{19}+38\,X^{18}+63\,X^{17}+81\,X^{16}+51\,X^{15}+67\,X^{14}+85\,X^{13}+83\,X^{12}+74\,X^{11}+82\,X^{10}+79\,X^9+74\,X^8+47\,X^7+46\,X^6+16\,X^5+34\,X^4+24\,X^3+62\,X^2+60\,X+22.$$

Et, une fois posé

$$\mathbb{Y}=\mathbb{X}[Y]/(Y^2-(X^3+12X+79)),\ \mathbb{X}=\mathbb{F}_q[X]/(f_7(X)),$$

le point P de coordonnées $(X,Y)\in\mathbb{Y}\times\mathbb{Y}$ est un point générique de 7-torsion. Pour s'en convaincre, on peut vérifier que $E[7]\simeq\mathbb{Z}/7\mathbb{Z}\times\mathbb{Z}/7\mathbb{Z}$ en posant

$$Q=(X^{97},Y^{97})=(92\,X^{23}+\cdots+84,Y(44\,X^{23}+\cdots+1)),$$

et en vérifiant que $E[7]=\{iP+jQ,\ (i,j)\in\{0,6\}^2\}$.

- Exemple de points de p-torsion. Soit $\mathbb{F}_{5^3}\simeq\mathbb{F}_5[t]/(t^3+t+1)$ et $E/\mathbb{F}_{5^3}\ : Y^2=X^3+X+t$. Un résultat classique prévoit que $E[p^i]$ est isomorphe à $\{O_E\}$ (E est dite supersingulière), ou à $\mathbb{Z}/p^i\mathbb{Z}$ sinon. Ici, on calcule $f_5(X)=2\,X^{10}+4\,tX^5+4\,t^2+t+1$. Ce qui, une fois les coefficients élevés à la puissance 5^2, conduit à,

$$f_5(X)=2(X^2+\left(2\,t^2+t+3\right)X+3\,t)^5.$$

On peut cependant, sans calculer $f_p(X)$ (qui est de degré $p(p-1)/2$ trouver directement le facteur $h_p(X)$ de degré $(p-1)/2$ en utilisant le résultat suivant [60].

Théorème 7. *Soit $E/\mathbb{F}_q\ : Y^2=X^3+a_4X+a_6$. Soient $r=(p-1)/2$ et $\gamma=\sqrt[p-1]{H_E}$, alors, on a $x\in\mathbb{F}_q(\gamma)$ pour $(x,y)\in E[p]$. Soient $\alpha_\nu=\nu(\nu-1)a_6$, $\beta_\nu=\nu(\nu-1/2)a_4$ et $\delta_\nu=\nu(\nu+1/2)$ pour $\nu=1,2,\ldots,r$. Si Δ_0 et Δ_1 sont les déterminants $r\times r$ et $(r-1)\times(r-1)$ définis par*

$$\Delta_0=\begin{vmatrix}\beta_1&\alpha_2&0&0&\cdots&0\\ \delta_1&-\gamma^2&\beta_3&\alpha_4&\ddots&\vdots\\ 0&\delta_2&-\gamma^2&\beta_4&\ddots&\vdots\\ \vdots&\ddots&\delta_3&-\gamma^2&\ddots&\alpha_r\\ \vdots&&\ddots&\ddots&\ddots&\beta_r\\ 0&\cdots&0&\delta_{r-1}&-\gamma^2&\end{vmatrix},\ \Delta_1=\begin{vmatrix}-\gamma^2&\beta_3&\alpha_4&\cdots&0\\ \delta_2&-\gamma^2&\beta_4&\ddots&\vdots\\ 0&\delta_3&-\gamma^2&\ddots&\alpha_r\\ \vdots&\ddots&\ddots&\ddots&\beta_r\\ 0&\cdots&0&\delta_{r-1}&-\gamma^2\end{vmatrix},\ \text{alors } x^p=\frac{\Delta_0^2-b\gamma^2\Delta_1^2}{4\gamma^2}.$$

Dans ce théorème, H_E est l'invariant de Hasse d'une courbe elliptique $E/\mathbb{K}$. Il est égal au coefficient de X^{p-1} du polynôme $(X^3+a_2X^2+a_4X+a_6)^{\frac{p-1}{2}}$. On a $H_E=0$ pour une courbe supersingulière.

Dans notre exemple, $(X^3+X+t)^2=X^6+2\,X^4+2\,tX^3+X^2+2\,tX+t^2$, et donc $H_E=2$. Comme X^4-H_E est irréductible dans $\mathbb{F}_{5^3}$, posons $\mathbb{K}=\mathbb{F}_{5^3}[\gamma]/(\gamma^4-2)$. Nous avons alors dans $\mathbb{K}$,

$$\Delta_0=\begin{vmatrix}3&2\,t\\ 4&4\,\gamma^2\end{vmatrix}=2\,\gamma^2+2\,t \text{ et } \Delta_1=\left|4\gamma^2\right|=4\gamma^2.$$

D'où, $x=\left(3\,t^2+t+1\right)\gamma^2+4\,t^2+2\,t+1$, et $P=\left(x,\left(t^2+2\,t+2\right)\gamma^3+\left(3\,t^2+4\,t+4\right)\gamma\right)\in E[5]$. Alors,

$$2P\ =\ \left(\left(2\,t^2+4\,t+4\right)\gamma^2+4\,t^2+2\,t+1,\ \left(3\,t^2+t+1\right)\gamma^3+\left(t^2+3\,t+3\right)\gamma\right),$$

et on retrouve,

$$h_5(X)=(X-X_P)(X-X_{2P})=X^2+\left(2\,t^2+t+3\right)X+3\,t.$$

1.3.3 Isogénies

- Propriétés. Pour les courbes elliptiques, une isogénie $\mathcal{I}$ est un morphisme de groupes algébriques envoyant O_{E_a} en O_{E_b}.

Théorème 8. *Soit $\mathcal{I}$ une isogénie non-nulle définie d'une courbe E_a vers une courbe E_b sur un corps $\mathbb{K}$. Alors* $\mathrm{Ker}(\mathcal{I})$ *est un groupe fini et si $\mathbb{K}$ est un corps algébriquement clos, $\mathcal{I}$ est surjective.*

Lorsque $\mathcal{I}$ est surjective, elle induit l'injection suivante sur les corps de fonctions associés à E_a et E_b, $\mathcal{I}^* : \mathbb{K}(E_b) \to \mathbb{K}(E_a), f \mapsto f \circ \mathcal{I}$. Les propriétés de séparabilité ou d'inséparabilité s'en déduisent.

Définition 3. *Si $\mathbb{K}$ est un corps algébriquement clos, une isogénie $\mathcal{I}$ entre deux courbes elliptiques E_a et E_b définies sur $\mathbb{K}$ est dite séparable ou inséparable si l'extension $\mathbb{K}(E_a)/\mathcal{I}^*(\mathbb{K}(E_b))$ a ces propriétés. On note alors* $\deg_s(\mathcal{I})$ *et* $\deg_i(\mathcal{I})$ *les degrés correspondants. Le degré de $\mathcal{I}$ est alors égal à* $\deg(\mathcal{I}) = \deg_s(\mathcal{I})\deg_i(\mathcal{I})$.

Le degré de séparabilité d'une isogénie est directement relié à son noyau.

Théorème 9. *Soit $\mathcal{I}$ une isogénie non-constante d'une courbe E_a vers une courbe E_b sur un corps algébriquement clos $\mathbb{K}$. Alors :*

1. *pour tout point S de $E_b(\mathbb{K})$, nous notons $\mathcal{I}^{-1}(S)$ le sous-ensemble de $E_a(\mathbb{K})$ qui a pour image S. Le cardinal de $\mathcal{I}^{-1}(S)$ est fini et égal à* $\deg_s(\mathcal{I})$ *;*
2. *si ℓ est un entier positif, $[\ell]_{E_a}$ est une isogénie de degré ℓ^2.*

Lorsque le corps $\mathbb{K}$ n'est pas algébriquement clos, le degré "s'entend" sur la clôture algébrique.

Théorème 10. *Soit $\mathcal{I}$ une isogénie non-constante définie d'une courbe E_a vers une courbe E_b sur un corps $\mathbb{K}$. Alors, il existe une unique isogénie $\hat{\mathcal{I}}$ de E_b vers E_a telle que $\hat{\mathcal{I}} \circ \mathcal{I} = [\deg(\mathcal{I})]_{E_a}$. De plus* $\deg(\hat{\mathcal{I}}) = \deg(\mathcal{I})$. *Cette isogénie est appelée* l'isogénie duale *de $\mathcal{I}$.*

Lorsqu'il existe une isogénie de degré $\ell \neq 0$ entre deux courbes elliptiques, ces courbes sont dites isogènes de degré ℓ. Sur $\mathbb{F}_q$, elles ont même cardinalité. La réciproque est vraie. La multiplication par un entier ℓ premier avec la caractéristique est l'exemple typique d'isogénie séparable, $[\ell]_E : E(\bar{\mathbb{F}}_q) \longrightarrow E(\bar{\mathbb{F}}_q), (X,Y) \longmapsto \ell(X,Y)$. On a $\deg_s([\ell]_E) = \ell^2$ (car $\#E[\ell] = \ell^2$) et $\deg_i([\ell]_E) = 1$, donc $\deg([\ell]_E) = \ell^2$. À l'inverse, le "petit" Frobenius, $\varphi_E : E(\bar{\mathbb{F}}_q) \longrightarrow E^\sigma(\bar{\mathbb{F}}_q), (X,Y) \longmapsto (X^p, Y^p)$, est inséparable. Il conduit classiquement à l'endomorphisme Frobenius $\phi_E = \varphi_E^n$ si $q = p^n$. On a $\deg_s(\varphi_E) = 1$ et $\deg_i(\varphi_E) = p$, donc $\deg(\varphi_E) = p$ (resp. $\deg(\phi_E) = q$).

Pour un endomorphisme $\mathcal{I}$, on a $\mathcal{I}^2 - (\mathcal{I} + \hat{\mathcal{I}})\mathcal{I} + \mathcal{I}\hat{\mathcal{I}} = 0$. En particulier, pour $\mathcal{I} = \Phi_E$, on retrouve $\phi_E^2 - [c]_E\phi_E + [q]_E = 0$.

- Formules de Vélu. Soit F, un sous-groupe fini de $E_a(\bar{\mathbb{K}})$, les formules de Vélu exhibe une isogénie séparable $\mathcal{I}$ ayant pour noyau F. Cette isogénie est donnée par les expressions suivantes :

$$E_a(\mathbb{K}) \to E_b(\mathbb{K})$$
$$P \mapsto \begin{cases} O_{E_b} & \text{si } P = O_{E_a}, \\ \left(X_P + \sum_{Q \in F - \{O_{E_a}\}} X_{P+Q} - X_Q \, , \, Y_P + \sum_{Q \in F - \{O_{E_a}\}} Y_{P+Q} - Y_Q \right) & \text{si } P = (X_P, Y_P). \end{cases}$$

Le théorème suivant explicite complètement cette construction. Il est du a Vélu.

Théorème 11. *Soient E_a une courbe définie sur un corps $\mathbb{K}$ algébriquement clos, F un sous-groupe fini de $E_a(\mathbb{K})$, $R \subset F$ tel que $F - E_a[2] = R \cup (-R)$ avec $R \cap -R = \emptyset$ et $S = F \cap E_a[2] - \{O_{E_a}\}$. Soient $\forall Q = (X_Q, Y_Q) \in F - \{O_{E_a}\}$,*

$$g_Q^r = 3X_Q^2 + 2a_2X_Q + a_4 - a_1Y_Q, \; g_Q^y = -2Y_Q - a_1X_Q - a_3,$$
$$t_Q = \begin{cases} g_Q^r & \text{si } Q \in S, \\ 2g_Q^r - a_1g_Q^y = 6X_Q^2 + d_2X_Q + d_4 & \text{sinon}, \end{cases}$$
$$u_Q = (g_Q^y)^2 = 4X_Q^3 + d_2X_Q^2 + 2d_4X_Q + d_6$$

et $t = \sum_{Q\in R\cup S} t_Q$, $w = \sum_{Q\in R\cup S} u_Q + X_Q t_Q$. *Alors, la correspondance* $E_a(\mathbb{K}) \to E_b(\mathbb{K})$, $(X_P, Y_P) \to (X_{\mathcal{I}(P)}, Y_{\mathcal{I}(P)})$, *où* E_b *est définie sur* $\mathbb{K}$ *par* $Y^2 + b_1XY + b_3Y = X^3 + b_2X^2 + b_4X + b_6$, *avec* $b_1 = a_1$, $b_2 = a_2$, $b_3 = a_3$, $b_4 = a_4 - 5t$ *et* $b_6 = a_6 - d_2t - 7w$ *et* $(X_{\mathcal{I}(P)}, Y_{\mathcal{I}(P)})$ *est définie par*

$$\left\{\begin{array}{rcl} X_{\mathcal{I}(P)} & = & X_P + \sum_{Q\in R\cup S}\left(\frac{t_Q}{X_P - X_Q} + \frac{u_Q}{(X_P-X_Q)^2}\right), \\ Y_{\mathcal{I}(P)} & = & Y_P + \sum_{Q\in R\cup S}\left(u_Q\frac{2Y_P + a_1X_P + a_3}{(X_P-X_Q)^3}\right. \\ & & \left. + t_Q\frac{a_1(X_P-X_Q)+Y-Y_Q}{(X_P-X_Q)^2} + \frac{a_1u_Q - g_Q^r g_Q^y}{(X_P-X_Q)^2}\right), \end{array}\right.$$

est une isogénie de degré $\#F$.

Il n'est pas difficile d'obtenir une version symétrique des formules de Vélu qui ont pour point de départ le polynôme

$$h(X) = \prod_{Q\in R}(X - X_Q) = X^d - h_1X^{d-1} + h_2X^{d-2} - h_3X^{d-3} + \cdots + (-1)^d h_0,$$

et qui conduisent, via des calculs dans $\mathbb{K}[X]$, à une expression rationnelle de la courbe isogène et de l'isogénie associée, en particulier pour ℓ impair. Les isogénies de degré ℓ obtenues sont de la forme

$$\begin{array}{rccl} \mathcal{I}: & E_a(\bar{\mathbb{F}}_q) & \longrightarrow & E_b(\bar{\mathbb{F}}_q), \\ & (X,Y) & \longmapsto & \left(\frac{g_\ell(X)}{h_\ell^2(X)}, \frac{l_\ell(X) + Yk_\ell(X)}{h_\ell^3(X)}\right) \end{array}$$

avec $g_\ell(X)$, $l_\ell(X)$, $k_\ell(X)$ et $h_\ell(X)$ des polynômes de degrés respectifs ℓ, $3(\ell-1)/2$, $(\ell-1)$ et $(\ell-1)/2$.

Exemple d'isogénie duale de φ_E. Pour $E/\mathbb{F}_{5^3} : Y^2 = X^3 + X + t$, avec $\mathbb{F}_{5^3} \simeq \mathbb{F}_5[t]/(t^3 + t + 1)$, on a $h_5(X) = X^2 + \left(2\,t^2 + t + 3\right)X + 3\,t$. On trouve donc,

$$\hat{\varphi_E}(X,Y) = \left(\frac{4\,x^5 + tx^4 + \left(2\,t^2+1\right)x^3 + 3\,tx^2 + \left(4\,t^2 + 4\,t + 3\right)x + 4\,t^2 + t + 3}{(x^2 + 2\,tx + 2\,t^2 + 3\,t + 3)^2}\right.,$$
$$\left. Y\frac{2\,x^6 + 2\,tx^5 + \left(t^2+2\right)x^4 + \left(t^2 + 3\,t + 4\right)x^3 + (t+1)\,x^2 + \left(2\,t^2 + 2\,t + 2\right)x + 4\,t^2 + t}{(x^2 + 2\,tx + 2\,t^2 + 3\,t + 3)^3}\right).$$

On peut remarquer que, comme $\deg_i \varphi_E = p$, $\deg_s \varphi_E = 1$ et $\deg_i \hat{\varphi_E} = 1$, $\deg_s \hat{\varphi_E} = p$, on a $\deg_i [p]_E = p$, $\deg_s [p]_E = p$ et que l'on retrouve bien $\deg [p]_E = p^2$.

2 Algorithmes

L'algorithme déterministe et polynomial en temps de Schoof [152, 153] qui permet de calculer la cardinalité d'une courbe elliptique en $O(\log^{5+\varepsilon} q)$ opérations élémentaires a d'abord été grandement amélioré par Atkin [5, 6] puis Elkies [45, 46] pour déboucher sur un algorithme cette fois-ci probabiliste mais, pour des corps finis de grande caractéristique, de complexité en moyenne $O(\log^{4+\varepsilon} q)$. Dans les corps finis de petite caractéristique, on doit à Couveignes [37] d'avoir levé les obstructions et ainsi d'avoir un algorithme de même complexité.

Dans l'algorithme tel que le décrit Schoof [152], on voit l'équation caractéristique du Frobenius,

$$\phi^2 - [c] \circ \phi + [q] = 0,$$

comme agissant linéairement sur le $\mathbb{F}_\ell$-espace vectoriel $E[\ell]$ (pour des entiers ℓ premiers entre eux) afin d'obtenir $c \bmod \ell$. Lorsque suffisamment de $c \bmod \ell$ sont connus, il suffit d'utiliser le théorème chinois pour remonter à la cardinalité de la courbe.

La complexité des calculs est directement reliée au degré de l'extension que l'on utilise pour manipuler la ℓ-torsion. L'algorithme de Schoof utilise les polynôme de ℓ-division $f_\ell(X)$, ce degré est donc de l'ordre

de $\ell^2/2$. Avec une arithmétique asymptotiquement rapide, on trouve une complexité égale à $O(\log^{5+\varepsilon} q)$ en temps et $O(\log^2 q)$ en espace. Une toute première implantation de l'algorithme pour des corps de caractéristique deux est due à Menezes, Vanstone et Zuccherato [123].

Cependant, pour $\ell \neq p$ et si l'idéal principal (ℓ) de décompose dans $\mathbb{Q}(\sqrt{c^2-4q})$, il existe en général deux isogénies de degré ℓ associées à deux facteurs $h_\ell(X)$ de degré $(\ell-1)/2$ de $f_\ell(X)$. Trouver ces isogénies à partir des $h_\ell(X)$ est faisable par les formules de Vélu mais cela nécessite la factorisation de $f_\ell(X)$ (et est donc inefficace). Atkin et Elkies ont montré que l'on peut faire l'inverse : trouver les isogénies rationnelles de degré ℓ définies à partir de E afin d'en déduire, sans factorisation, le polynôme $h_\ell(X)$. La complexité en temps décroît alors jusqu'à $O(\log^{4+\varepsilon} q)$.

En fait, quatre algorithmes sont connus pour calculer explicitement des isogénies de degré ℓ sur $\mathbb{F}_{p^n}$.

Elkies/Atkin (1992) : valide pour $p \gg \ell$, il nécessite $O(\ell^2)$ multiplications dans $\mathbb{F}_{p^n}$ et le stockage de $O(\ell)$ éléments.

Couveignes I (1994) : le premier algorithme connu pour $p \ll \ell$, il nécessite des calculs dans les groupes formels définis par E_a et E_b ; il nécessite $O(\ell^3)$ multiplications et un stockage en $O(\ell^2)$.

Lercier (1996) : seulement pour $p = 2$, on se débarrasse des groupes formels et on obtient un algorithme de complexité heuristique $O(\ell^3)$ et un stockage en $O(\ell^2)$. C'est probablement l'algorithme le plus efficace en pratique pour les degrés ℓ nécessaires à l'algorithme `SEA`.

Couveignes II (1997) : pour $p \ll \ell$, l'algorithme correspondant est de complexité $O(\ell^{2+\epsilon})$ en temps et $O(\ell)$ en espace. Il nécessiterait, s'il était implanté, des algorithmes complexes à base de FFT.

Pour $\ell = p$, notons enfin que l'on est capable d'exhiber un facteur de degré $(p-1)/2$ avec les idées de Gunji.

2.1 Courbes isogènes

Il existe un polynôme $\Phi_\ell(X,Y)$ appelé "polynôme modulaire" qui caractérise complètement les courbes isogènes de degré ℓ. Ainsi, pour une courbe E de $\mathbb{F}_q$, les racines de $\Phi_\ell(j_E, Y)$ sont les invariants de courbes isogènes de degré ℓ à E. Le polynôme $\Phi_\ell(X,Y)$ est à coefficients dans $\mathbb{Z}$. Il est calculé à l'aide d'invariants de courbes définies sur $\mathbb{C}$ par un réseau $\mathbb{C}/(\omega_1\mathbb{Z} + \omega_2\mathbb{Z})$. Malheureusement, la taille de ses coefficients augmentent vite avec ℓ. Le problème a été résolu par Atkin avec l'introduction de variantes de ces polynômes, les polynômes $\Phi^c_\ell(X,Y)$ et $\Phi^*_\ell(X,Y)$.

Théorème 12. *Soient E une courbe elliptique définie sur $\mathbb{F}_q$ et $\mathbb{E}$ un sous-groupe d'ordre ℓ de $E[\ell]$. Alors le degré r de la plus petite extension sur laquelle on peut définir l'isogénie quotient par $\mathbb{E}$ est égal au plus petit entier ρ tel que $\phi^\rho_{E[\ell]}(\mathbb{E}) = \mathbb{E}$.*

Théorème 13. *Soit E une courbe non-supersingulière de $\mathbb{F}_q$ à j-invariant distinct de 0 et 1728. Soit $\gamma_1(Y)\cdots\gamma_s(Y)$ la factorisation sur $\mathbb{F}_q[Y]$ de $\Phi_\ell(j_E, Y)$ en polynômes irréductibles. Alors nous avons trois possibilités pour les degrés des polynômes $\gamma_1, \gamma_2, \ldots, \gamma_s$:*

(i) $\begin{cases} (a)\ 1, \ell \\ (b)\ 1, 1, 1, \ldots, 1 \end{cases}$ *si $c^2 - 4q = 0 \bmod \ell$.*

(ii) $1, 1, r, \ldots, r$ *si $c^2 - 4q$ est un carré dans $\mathbb{F}^*_\ell$.*

(iii) $r, \ldots, r$ *avec $r > 1$ si $c^2 - 4q$ n'est pas un carré dans $\mathbb{F}_\ell$.*

Dans tous les cas, r est le plus petit entier tel que

$$\forall P \in E[\ell], \exists \kappa \in \mathbb{N}, \phi^r(P) = \kappa P.$$

- Exemple de courbes isogènes. Notons $\overline{\gamma_0 + \gamma_1 2 + \ldots + \gamma_7 2^7} = \gamma_0 + \gamma_1 t + \ldots + \gamma_7 t^7$. Soit $E_{\overline{7}}$, la courbe elliptique E_a définie sur $\mathbb{F}_{2^8} \simeq \mathbb{F}_2[t]/(t^8 + t^4 + t^3 + t + 1)$ par $a_6 = \overline{7}$, c'est-à-dire la courbe

$$E_{\overline{7}} : Y^2 + XY = X^3 + \overline{7}.$$

On a $\Phi^c_5(X,Y) \equiv X^6 + X^4 + X^2 + XY + \overline{1} \bmod 2$. D'où

$$\Phi^c_5(X, 1/\overline{7}) = (X + \overline{52})(X + \overline{130}) \times (X^2 + \overline{73}X + \overline{195}) \times (X^2 + \overline{255}X + \overline{239}).$$

Et $\Phi^c_5(\overline{52}, Y) = Y + 1/\overline{8}$. On a donc une isogénie de degré 5 entre $E_{\overline{7}}$ et $E_{\overline{8}}$.

2.2 Calcul d'isogénie en grande caractéristique

Sur $\mathbb{C}$, une isogénie entre $E_a\ : Y^2 = X^3 + a_4X + a_6$ et $E_b\ : Y^2 = X^3 + b_4X + b_6$ s'exprime en terme de réseau comme

$$\begin{array}{ccc} \mathbb{C}/(\omega_1\mathbb{Z} + \omega_2\mathbb{Z}) & \to & \mathbb{C}/(\frac{\omega_1}{\ell}\mathbb{Z} + \omega_2\mathbb{Z}), \\ z & \mapsto & z. \end{array}$$

Soient $\wp_a(z)$ et $\wp_b(z)$ les fonctions de Weierstrass associées à E_a et E_b. Nous avons $\forall z \in \mathbb{C}$,

$$\wp_a(z) = \frac{1}{z^2} + \sum_{k=1}^{\infty} c_{a_k} z^{2k} \text{ et } \wp_b(z) = \frac{1}{z^2} + \sum_{k=1}^{\infty} c_{b_k} z^{2k}$$

avec $c_{a_1} = -\frac{a_4}{5},\ c_{a_2} = -\frac{a_6}{7},\ c_{a_k} = \dfrac{3}{(k-2)(2k+3)} \displaystyle\sum_{j=1}^{k-2} c_{a_j} c_{a_{k-1-j}}$ pour $k \geqslant 3$,

et de même $c_{b_1} = -\frac{b_4}{5},\ c_{b_2} = -\frac{b_6}{7},\ c_{b_k} = \dfrac{3}{(k-2)(2k+3)} \displaystyle\sum_{j=1}^{k-2} c_{b_j} c_{b_{k-1-j}}$ pour $k \geqslant 3$.

Notons $I(X) = g(X)/h^2(X)$ l'abscisse de l'image par $\mathcal{I}$ d'un point (X, Y) de $E_a(\mathbb{C})$. Nous avons,

$$\wp_b(z) = I(\wp_a(z)).$$

Ce qui suffit pour déterminer $\mathcal{I}$.

2.3 Isogénies pour $\mathbb{F}_{p^n}$, p petit : premières idées de Couveignes

Nous devons à Couveignes [37] le premier algorithme utilisable pour les corps de petites caractéristiques. Il consiste à remplacer la paramétrisation de Weierstrass par celle, utilisable pour tout corps fini, des groupes formels où X est développé en série de Laurent en un paramètre t et $Y = -X/t$.

On cherche sur $\mathbb{F}_{2^n}$, une isogénie $\mathcal{I}$ de degré ℓ entre deux courbes isogènes $E_a\ : Y^2 + XY = X^3 + a_6$ et $E_b\ : Y^2 + XY = X^3 + b_6$ En "réécrivant" X et Y comme des séries en $t = -X/Y$, on a avec $s(t) = -1/Y$, $t^3 + ts(t) + a_6 s(t)^3 = s(t)$, et la détermination de $\mathcal{I}$ est équivalente à celle d'une série

$$U(t) = \frac{g_\ell(X(t))}{h_\ell^2(X(t))} = t + \sum_{i=2}^{\infty} u_i t^i.$$

2.3.1 Groupe formel

Les points formels $(t(\zeta), s(\zeta))$ définissent un groupe. Les formules d'additions sont $(t_3(\zeta), s_3(\zeta)) = (t_1(\zeta), s_1(\zeta)) \oplus (t_2(\zeta), s_2(\zeta))$, avec

$$\begin{cases} \lambda = \dfrac{s_1 + s_2}{t_1 + t_2}, \qquad \nu = s_1 + \lambda t_1, \\ t_3 = \dfrac{t_1 + t_2 + \lambda + a_6\lambda^2(s_1 + s_2 + \nu)}{1 + t_1 + t_2 + \lambda + a_6\lambda^2\,(s_1 + s_2 + \nu + \lambda)},\ s_3 = \nu + (\lambda + \nu)t_3, \end{cases}$$

et $(t_3(\zeta), s_3(\zeta)) = [2](t(\zeta), s(\zeta))$, avec

$$\begin{cases} \lambda = \dfrac{s + \sqrt{a_6}s^2}{t^2}, \qquad \nu = \dfrac{s}{t}, \qquad \rho = \dfrac{\lambda + \sqrt{a_6}\nu\lambda^2}{1 + \lambda + \sqrt{a_6}\nu\lambda^2 + \sqrt{a_6}\lambda^3}, \\ t_3 = \rho^2,\ s_3 = (\nu + (\lambda + \nu)\rho)^2\,. \end{cases}$$

2.3.2 Détermination de $U(t)$

On part de $\forall (t_1, t_2),\ U(t_1 \oplus t_2) = U(t_1) \oplus U(t_2)$. Ainsi :

- lorsque $i \neq 2^k$, u_i est déterminé de façon unique connaissant $u_2, \ldots, u_{i-1}$ par $U(t \oplus At) = U(t) \oplus U(At)$ avec A choisi tel que $(1 + A)^i \neq 1 + A^i$.
- lorsque $i = 2^k$, u_i vérifie une équation du second degré déduite de $U([2](t)) = [2](U(t))$. On a ainsi deux valeurs possibles pour u_i.

2.3.3 Comment trouver $\mathcal{I}$ à partir de U

Parmi les morphismes de groupes formels $[N]\circ U(t)$ pour les entiers N impairs $1 \leqslant N \leqslant 2^k$ avec $2^{k-1} \leqslant 4\ell \leqslant 2^k$, un morphisme correspond à l'isogénie $\mathcal{I}(t)$ recherchée. On teste donc si les 4ℓ termes de $[N]\circ U(t)$ correspondent à une fraction rationnelle de degré ℓ en X avec l'algorithme de Massey-Berlekamp. Quand c'est le cas, la reconstruction de $\mathcal{I}$ en fonction de X conduit au facteur $h_\ell(X)$ de $f_\ell(X)$.

- Exemple. Dans $E_{\overline{7}} : Y^2 + XY = X^3 + \overline{7}$ définie sur $\mathbb{F}_{2^8}$ comme précédemment, on cherche une isogénie de degré $\ell = 5$ entre $E_{\overline{7}}$ et $E_{\overline{8}}$. Le point

$$(t(\zeta), s(\zeta)) = \left(\begin{array}{l} \zeta, \quad \zeta^3 + \zeta^4 + \zeta^5 + \zeta^6 + \zeta^7 + \zeta^8 + \overline{6}\zeta^9 + \zeta^{10} + \zeta^{11} + \zeta^{12} + \overline{6}\zeta^{13} + \zeta^{14} + \\ \qquad \overline{20}\zeta^{15} + \overline{20}\zeta^{16} + \overline{6}\zeta^{17} + \zeta^{18} + \zeta^{19} + \zeta^{20} + \overline{6}\zeta^{21} + \zeta^{22} + \overline{20}\zeta^{23} \end{array} \right)$$

est alors un élément du groupe formel défini par $t(\zeta)^3 + t(\zeta)s(\zeta) + \overline{7}s(\zeta)^3 = s(\zeta)$. Soit

$$U(\zeta) = \zeta + u_2\zeta^2 + u_3\zeta^3 + u_4\zeta^4 + u_5\zeta^5 \ldots$$

Avec $A = \overline{2}$,

$$U(\zeta \oplus A\zeta) + U(\zeta) \oplus U(A\zeta) = \left(\overline{6}\, u_2 + \overline{6}\, u_3\right)\zeta^3 + \left(\overline{4}\, {u_2}^2 + \overline{4}\, u_2\right)\zeta^4 + \ldots$$

et

$$U([2]\zeta) + [2](U(\zeta)) = \left({u_2}^2 + u_2\right)\zeta^4 + \left({u_3}^2 + u_3\right)\zeta^6 + \left(u_4 + {u_4}^2 + \overline{15}\right)\zeta^8 + \ldots$$

Par conséquent,

$$\begin{array}{rcll} u_2^2 + u_2 = 0 & \Rightarrow & u_2 \in \{\overline{0}, \overline{1}\}, & u_2 = 0, \\ \overline{6}\, u_2 + \overline{6}\, u_3 = 0 & \Rightarrow & u_3 \in \{\overline{0}\}, & u_3 = 0, \\ u_4 + {u_4}^2 + \overline{15} = 0 & \Rightarrow & u_4 \in \{\overline{56}, \overline{57}\}, & u_4 = \overline{56}, \\ \vdots & & \vdots & \vdots \end{array}$$

Ainsi, en réitérant le procédé,

$$\begin{array}{rcl} U(\zeta) & = & \zeta + \overline{56}\zeta^4 + \overline{56}\zeta^5 + \overline{15}\zeta^7 + \overline{16}\zeta^8 + \overline{31}\zeta^9 + \overline{219}\zeta^{10} + \overline{124}\zeta^{11} + \overline{5}\zeta^{12} + \overline{44}\zeta^{13} + \overline{91}\zeta^{14} + \\ & & \overline{47}\zeta^{15} + \overline{210}\zeta^{16} + \overline{201}\zeta^{17} + \overline{231}\zeta^{18} + \overline{198}\zeta^{19} + \overline{188}\zeta^{20} + \overline{118}\zeta^{21} \ldots \end{array}$$

Par additions dans le groupe formel,

$$\begin{array}{rcl} [2](U(\zeta)) & = & \zeta^2 + \overline{63}\zeta^8 + \overline{63}\zeta^{10} + \overline{29}\zeta^{14} + \overline{184}\zeta^{16} + \overline{165}\zeta^{18} + \overline{227}\zeta^{20} + \ldots \\ [3](U(\zeta)) & = & \zeta + \zeta^2 + \zeta^3 + \overline{56}\zeta^4 + \overline{56}\zeta^5 + \overline{56}\zeta^6 + \overline{55}\zeta^7 + \overline{39}\zeta^8 + \overline{39}\zeta^9 + \overline{244}\zeta^{10} + \overline{84}\zeta^{11} + \overline{154}\zeta^{12} + \\ & & \overline{28}\zeta^{13} + \overline{79}\zeta^{14} + \overline{52}\zeta^{15} + \overline{247}\zeta^{16} + \overline{51}\zeta^{17} + \overline{44}\zeta^{18} + \overline{66}\zeta^{19} + \overline{102}\zeta^{20} + \overline{84}\zeta^{21} + \ldots \end{array}$$

Et avec l'algorithme de Massey-Berlekamp, on trouve finalement,

$$[3](U(\zeta)) = \mathcal{I}(X(\zeta)) = \frac{X^5(\zeta) + \overline{15}X^3(\zeta) + \overline{140}X(\zeta)}{(X^2(\zeta) + \overline{57}X(\zeta) + \overline{74})^2}.$$

2.4 Isogénies pour $\mathbb{F}_{p^n}$, p petit : secondes idées de Couveignes

Le deuxième visage des idées de Couveignes est un second algorithme qui consiste à étudier le comportement de l'isogénie au voisinage de "l'équivalent de $z = 0$" pour une courbe elliptique définie sur un corps fini $\mathbb{F}_q$ de caractéristique p, c'est-à-dire les points de p^k-torsion ($k \in \mathbb{N}^*$).

Au premier abord, cet algorithme semble assez simple puisqu'il s'agit ni plus ni moins d'une interpolation de l'isogénie sur les points de p^k-torsion des courbes isogènes [38]. Plus précisément, l'idée consiste alors à interpoler $\mathcal{I}$ sur $E_a[p^k]$ par

$$\mathcal{I}(iP_a) = iP_b \text{ pour } P_a \text{ et } P_b \text{ de } E_a[p^k] \text{ et } E_b[p^k].$$

Les calculs qui seraient mis en œuvre par une implantation naïve de ces idées conduiraient cependant à des coûts prohibitifs. Grâce notamment à une méthode ingénieuse de calcul des points de p^k-torsion

sur une courbe elliptique définie sur $\mathbb{F}_q$ initiée par Hasse et Witt et explicitée par Voloch [169], ces coûts peuvent être notablement réduits. L'idée principale est de définir ces points de p^k-torsion via des extensions d'Artin-Schreier.

En conjonction avec d'autres idées de Couveignes [39], la complexité asymptotique de cet algorithme est à p fixé de $O(n\ell^{1+\epsilon})$ opérations dans $\mathbb{F}_q$.

2.5 Isogénies pour $\mathbb{F}_{2^n}$: une approche spécifique

Les deux algorithmes de Couveignes décrits dans les paragraphes précédents sont bien sûr directement applicables à ce cas. Pourtant, contrairement aux corps finis de très grande caractéristique, il s'est avéré en pratique que le coût majeur dans notre implantation pour déterminer le nombre de points d'une courbe elliptique définie sur $\mathbb{F}_{2^n}$ est celui du calcul d'isogénie.

C'est pourquoi nous avons développé une autre méthode pour résoudre ce problème [103]. Basée sur des identités algébriques satisfaites par les isogénies, par exemple la commutativité avec la multiplication par deux sur la courbe, sa complexité asymptotique heuristique est similaire à celle du premier algorithme de Couveignes, quoique cette méthode soit nettement plus efficace en pratique.

Nous suivons une approche en trois temps. Tout d'abord on caractérise les isogénies définies sur $\mathbb{F}_{2^n}$. On donne ensuite les conditions qui sont vérifiées par ces isogénies et on en déduit la fraction rationnelle associée.

2.5.1 Résultat principal

Sur $\mathbb{F}_{2^n}$, on recherche une isogénie séparable $\mathcal{I}$ de degré ℓ entre deux courbes isogènes $E_a : Y^2+XY = X^3+a$ et $E_b : Y^2+XY = X^3+b$. On peut alors montrer que $\mathcal{I}$ qui est donnée par

$$\begin{array}{rccc} \mathcal{I}: & E_a(\mathbb{F}_{2^n}) & \longrightarrow & E_b(\mathbb{F}_{2^n}), \\ & (X_P, Y_P) & \longmapsto & \left(\frac{G(X)}{Q^2(X)}, \frac{H(X)+YK(X)}{Q^3(X)}\right), \end{array}$$

où $Q(X), G(X), H(X)$ et $K(X)$ sont des polynômes de degré au plus d, ℓ, $3d$ et $2d$ où $d = (\ell-1)/2$ [103]. Alors,

1. $G(X) = XP^2(X)$ où $P(X)$ est donné par $\frac{\sqrt[8]{a}}{\sqrt[8]{b}}\left(\sqrt[4]{a}\right)^d P(X) = X^d Q(\sqrt{a}/X)$;
2. $K(X) = P^2(X)Q(X)$;
3. $H(X) = XR(X)P(X) + \sqrt{b}Q^3(X) + \sqrt{a}P^2(X)Q(X)$ avec $R(X) = X(PQ)'(X)$ ou $R(X) = (XPQ)'(X)$.

Pour démontrer ce résultat, on se sert du fait qu'il existe un unique point d'ordre 2 sur $E_a(\mathbb{F}_{2^n})$, $P_a = (0, \sqrt{a})$. On a alors $\mathcal{I}(P_a) = P_b$. où P_b est l'unique point d'ordre 2 de $E_b(\mathbb{F}_{2^n})$. Alors, on tire avantage de $\forall S = (X, Y) \in E_a(\mathbb{F}_{2^n})$, $\mathcal{I}(S + P_a) = \mathcal{I}(S) + P_b$, $\mathcal{I}(-S) = -\mathcal{I}(S)$ et $\mathcal{I}(S) \in E_b(\mathbb{F}_{2^n})$.

En utilisant de plus que pour tout point S de E_a, $\mathcal{I}([2](S)) = [2](\mathcal{I}(S))$, on déduit que,

$$\left(\frac{XP^2(X)}{Q^2(X)}\right) \circ \left(X^2 + \frac{a}{X^2}\right) = \left(X^2 + \frac{b}{X^2}\right) \circ \left(\frac{XP^2(X)}{Q^2(X)}\right).$$

D'où l'on exhibe deux équations suffisantes à la mise au point d'un algorithme de résolution,

$$\left(X + \sqrt[4]{a}\right) X^d \widehat{P}\left(X + \sqrt{a}/X\right) = XP^2(X) + \sqrt[4]{b}Q^2(X) \tag{1}$$

et

$$X^d \widehat{Q}(X + \sqrt{a}/X) = Q(X)P(X) \tag{2}$$

où $\widehat{P}(X) = \sqrt{P(X^2)}$ et $\widehat{Q}(X) = \sqrt{Q(X^2)}$.

- Corrolaires immédiats. Les équations précédentes permettent d'exprimer $Q(X)$ en fonction de $P(X)$ et, de plus, figent certains des coefficients de $P(X)$. Plus précisément, si l'on pose $P(X) = \sum_{i=0}^{d} p_i^2 X^i$, $Q(X) = X^d + \sum_{i=0}^{d-1} q_i^2 X^i$, $\alpha = \sqrt[4]{a}$ et $\beta = \sqrt[4]{b}$, alors on a

$$q_i = \frac{\sqrt[4]{\alpha}}{\sqrt[4]{\beta}} \sqrt{\alpha}^{d-2i} p_{d-i}, \ \forall i \in \{0, \ldots, d\},$$

et

$$p_d = 1, p_{d-1} = \alpha + \beta, p_{d-2} = \begin{cases} p_{d-1}^4 + \alpha p_{d-1} + \alpha^2 & \text{si } d \text{ est impair,} \\ p_{d-1}^4 + \alpha p_{d-1} & \text{si } d \text{ est pair,} \end{cases} \quad p_0 = \sqrt[4]{\alpha^{2d} + \alpha^{2d-1} p_{d-1}}.$$

2.5.2 Calculs

Les équations (1) et (2) sont suffisantes pour permettre le calcul des isogénies par la méthode des "coefficients indéterminés". Une façon cependant d'obtenir une complexité "décente" est de tirer partie de la forme très particulière du système obtenu. En fait, on explique ici sur un exemple comment se servir de l'équation (1), puis comment obtenir l'algorithme final en ajoutant les informations de l'équation (2).

On souhaite calculer une isogénie de degré 17 entre $E_{\overline{7}}$ et $E_{\overline{187}}$ dans $\mathbb{F}_2[t]/(t^8 + t^4 + t^3 + t + 1)$. Au début, $\alpha = \sqrt[4]{\overline{7}}$ et donc $\alpha = \overline{182}$, $\beta = \sqrt[4]{\overline{187}}$ d'où $\beta = \overline{10}$. Ainsi, $p_8 = \overline{1}$, $p_7 = \overline{188}$, $p_6 = \overline{245}$ et $p_0 = \overline{52}$. L'équation (1) conduit à un système semi-linéaire, ici,

$$\begin{cases} p_1^4 &= \overline{213}\, p_5 + \overline{246}, \\ p_2^4 &= \overline{205}\, p_5 + \overline{21}\, p_4 + \overline{40} + \overline{66}\, p_3, \\ p_3^4 &= \overline{115}\, p_1 + \overline{79} + \overline{47}\, p_3 + \overline{7}\, p_2, \\ p_4^4 &= \overline{131} + \overline{115}\, p_3 + \overline{182}\, p_1, \\ p_5^4 &= \overline{182}\, p_3 + \overline{25} + p_2 + \overline{115}\, p_5, \\ 0 &= 182\, p_5 + 241 + p_4. \end{cases}$$

Une fois substitués p_0 et p_d, p_{d-1}, p_{d-2} par leurs valeurs, on doit résoudre sur $\mathbb{F}_2$ un système linéaire de $n(d-2)$ équations avec $n(d-3)$ inconnues pour obtenir $p_1, \ldots, p_{d-3}$. Ainsi,

$$\begin{cases} p_5 &= \overline{219}\, p_1^4 + \overline{73}, \\ p_3 &= \overline{173} + \overline{55}\, p_2^4, \\ 0 &= \overline{161} + \overline{115}\, p_1 + \overline{195}\, p_2^{16} + \overline{156}\, p_2^4 + \overline{7}\, p_2, \\ 0 &= \overline{211}\, p_1^{16} + \overline{182}\, p_1 + \overline{36} + \overline{209}\, p_2^4, \\ 0 &= \overline{146}\, p_1^{16} + \overline{115} + \overline{139}\, p_2^4 + \overline{43}\, p_1^4 + p_2, \\ p_4 &= \overline{226} + \overline{53}\, p_1^4. \end{cases}$$

Cependant, résoudre tel quel est asymptotiquement trop complexe ($O(\ell^3 n^3)$ opérations seraient nécessaires). Même si l'équation (2) est non-linéaire, elle donne par contre des informations complémentaires,

$$\begin{cases} p_1^2 + \overline{163}\, p_1 + \overline{20} &= 0, \\ p_2^2 + \overline{67}\, p_2 + \overline{75}\, p_1^2 + \overline{81} &= 0. \end{cases}$$

Donc,

$$\begin{cases} p_1 &= \overline{88} + \overline{163}\, \pi_1, \\ p_2 &= \overline{134} + \overline{5}\, \pi_1 + \overline{67}\, \pi_2, \end{cases} \quad \text{avec } (\pi_1, \pi_2) \in \{0,1\}^2.$$

Une fois exprimés $p_1, \ldots, p_d$ en fonctions des variables binaires $\pi_1, \ldots, \pi_k$, on doit encore résoudre de l'ordre de $d/3$ équations pour trouver les valeurs prises par ces variables binaires. Ici,

$$\begin{cases} \overline{84}\, \pi_2 + \overline{254}\, \pi_1 + \overline{170} &= 0, \\ \overline{236}\, \pi_1 + \overline{191} + \overline{83}\, \pi_2 &= 0, \\ \overline{188}\, \pi_1 + \overline{243}\, \pi_2 + \overline{79} &= 0. \end{cases}$$

Donc, $\pi_1 = 1, \pi_2 = 1$, et $p_1 = \overline{251}, p_2 = \overline{192}, p_3 = \overline{151}, p_4 = \overline{3}, p_5 = \overline{211}$. Finalement,

$$Q(X) = X^8 + \overline{92}\, X^7 + \overline{88}\, X^6 + \overline{222}\, X^5 + \overline{40}\, X^4 + \overline{38}\, X^3 + \overline{135}\, X^2 + \overline{119}\, X + \overline{168}.$$

Remarque 5. *La clef de l'algorithme est d'observer qu'étant donné que chacune des équations du second degré du système (2) doit avoir une solution, les variables π_i sont reliées entre elles. Ainsi, dès que $K > 10$, chaque fois que l'on écrit p_K en fonction d'une nouvelle variable binaire π_{K-1}, nous observons que nous sommes capable de l'exprimer comme une fonction polynomiale des autres π_i excepté, un nombre logarithmique de fois. Le nombre de variables π_i réellement nécessaire est donc de l'ordre de $\log(\ell)$, on trouve ainsi une complexité totale en $O(\ell^3)$.*

2.6 Le théorème chinois avec "Chinese & Match"

Le dessein de l'algorithme `SEA` est d'obtenir un nombre minimal de candidats pour c modulo de petits nombres entiers ℓ premiers entre eux tels que

$$\prod_{\substack{\text{entiers premiers}\\ \text{entre eux } \ell}} \ell > 4\sqrt{q}. \tag{3}$$

Lorsque l'on a un unique candidat pour chaque ℓ, une application directe du théorème des restes chinois produit le nombre de points. Malheureusement, l'algorithme `SEA` produit un candidat unique pour $c \bmod \ell$ pour seulement la moitié des nombres premiers ℓ, en général pour les nombres d'Elkies et leurs puissances. Pour les nombres d'Atkin, nous avons un nombre pair de candidats. Une solution évidente serait de conserver seulement les nombres d'Elkies. Même si alors, un plus grand nombre d'entiers ℓ est nécessaire, la complexité asymptotique de l'algorithme reste identique. Mais dans la pratique, il est bien meilleur de considérer des entiers ℓ de plus petite taille en substituant à quelques entiers d'Elkies des entiers d'Atkin ℓ tel que le nombre de candidats pour $c \bmod \ell$ est le plus petit possible comparé à la taille de ℓ. Bien sûr, nous avons dans ce cas un nombre C de candidats pour le cardinalité que nous cherchons.

Un approche naïve pour trouver le nombre de points parmi les C candidats consiste à multiplier un point P de la courbe par chaque candidat jusqu'à ce que nous obtenons l'élément neutre. Une approche clairement plus efficace utilise la variante "pas de bébés et pas de géants" proposée par Atkin [5], appelée "Match & Sort". Sa complexité est de $O(C^{\frac{1}{2}})$ additions dans $E(\mathbb{F}_q)$, i.e une complexité de $O(C^{\frac{1}{2}} \log^2 q)$ opérations élémentaires. Le stockage nécessaire est de $O(C^{\frac{1}{2}} \log q)$ bits.

La méthode appelée "Chinese & Match" que nous avons développée en collaboration avec Joux suit une approche différente [75], en particulier, nous nous débarrassons des additions sur courbes elliptiques. Elle profite du fait que nous avons pour de grand calculs, comme l'exemple de $\mathbb{F}_{2^{1663}}$ qui nous sert de fil conducteur ici, beaucoup plus d'information sur c que ce qui est utilisé dans l'approche d'Atkin. Au lieu de chercher une égalité de points, l'idée est d'éliminer les mauvais candidats en vérifiant que ces derniers ne satisfont pas l'ensemble des congruences produites par l'algorithme `SEA`.

2.6.1 Un exemple

Dans, $\mathbb{F}_{2^{1663}} = \mathbb{F}_2[t]/(t^{1663}+t^9+t^8+t^6+t^4+t^3+1)$, nous considérons la courbe $E : Y^2+XY = X^3+a_6$, où $a_6 = t^{16} + t^{14} + t^{13} + t^9 + t^8 + t^7 + t^6 + t^5 + t^4 + t^3$. Alors $\#E(\mathbb{F}_{2^{1663}}) = 2^{1663} + 1 - c$ avec

$$c \in [-2\sqrt{2^{1663}}, 2\sqrt{2^{1663}}] \simeq [-4 \times 10^{250}, 4 \times 10^{250}].$$

Les congruences produites modulo l'ensemble $\mathfrak{U}$ des entiers d'Elkies sont les suivantes.

$\mathcal{T}_{2^{12}} = \{2561\}$, $\mathcal{T}_{3^7} = \{1566\}$, $\mathcal{T}_{5^2} = \{17\}$, $\mathcal{T}_{7^2} = \{2\}$, $\mathcal{T}_{11^3} = \{495\}$, $\mathcal{T}_{13} = \{7\}$,
$\mathcal{T}_{17^2} = \{0\}$, $\mathcal{T}_{19} = \{12\}$, $\mathcal{T}_{23^2} = \{72\}$, $\mathcal{T}_{29} = \{12\}$, $\mathcal{T}_{31} = \{29\}$, $\mathcal{T}_{37^2} = \{357\}$,
$\mathcal{T}_{41} = \{1\}$, $\mathcal{T}_{43^2} = \{301\}$, $\mathcal{T}_{47^2} = \{599\}$, $\mathcal{T}_{53^2} = \{1284\}$, $\mathcal{T}_{59} = \{22\}$, $\mathcal{T}_{67^2} = \{829\}$,
$\mathcal{T}_{73} = \{43\}$, $\mathcal{T}_{79} = \{38\}$, $\mathcal{T}_{89^2} = \{7554\}$, $\mathcal{T}_{101} = \{54\}$, $\mathcal{T}_{103} = \{57\}$, $\mathcal{T}_{109} = \{39\}$,
$\mathcal{T}_{127} = \{26\}$, $\mathcal{T}_{131} = \{5\}$, $\mathcal{T}_{151} = \{64\}$, $\mathcal{T}_{157} = \{110\}$, $\mathcal{T}_{163} = \{67\}$, $\mathcal{T}_{167} = \{0\}$,
$\mathcal{T}_{179} = \{162\}$, $\mathcal{T}_{223} = \{21\}$, $\mathcal{T}_{227} = \{113\}$, $\mathcal{T}_{229} = \{44\}$, $\mathcal{T}_{233} = \{86\}$, $\mathcal{T}_{251^2} = \{53764\}$,
$\mathcal{T}_{257} = \{146\}$, $\mathcal{T}_{263} = \{191\}$, $\mathcal{T}_{271} = \{70\}$, $\mathcal{T}_{293} = \{215\}$, $\mathcal{T}_{337} = \{20\}$, $\mathcal{T}_{347} = \{108\}$,
$\mathcal{T}_{373} = \{54\}$, $\mathcal{T}_{383} = \{214\}$, $\mathcal{T}_{401} = \{119\}$, $\mathcal{T}_{431} = \{313\}$, $\mathcal{T}_{433} = \{328\}$, $\mathcal{T}_{439} = \{76\}$,
$\mathcal{T}_{449} = \{367\}$, $\mathcal{T}_{461} = \{56\}$, $\mathcal{T}_{463} = \{427\}$, $\mathcal{T}_{467} = \{328\}$, $\mathcal{T}_{491} = \{398\}$, $\mathcal{T}_{503} = \{4\}$,
$\mathcal{T}_{509} = \{60\}$, $\mathcal{T}_{523} = \{369\}$, $\mathcal{T}_{547} = \{441\}$, $\mathcal{T}_{563} = \{355\}$, $\mathcal{T}_{569} = \{367\}$, $\mathcal{T}_{571} = \{171\}$,
$\mathcal{T}_{577} = \{76\}$, $\mathcal{T}_{587} = \{236\}$, $\mathcal{T}_{599} = \{471\}$, $\mathcal{T}_{613} = \{182\}$, $\mathcal{T}_{619} = \{310\}$, $\mathcal{T}_{631} = \{258\}$,
$\mathcal{T}_{643} = \{230\}$, $\mathcal{T}_{647} = \{148\}$, $\mathcal{T}_{677} = \{483\}$, $\mathcal{T}_{701} = \{299\}$, $\mathcal{T}_{709} = \{636\}$, $\mathcal{T}_{719} = \{310\}$,
$\mathcal{T}_{727} = \{597\}$, $\mathcal{T}_{797} = \{328\}$, $\mathcal{T}_{811} = \{632\}$, $\mathcal{T}_{853} = \{164\}$, $\mathcal{T}_{857} = \{98\}$, $\mathcal{T}_{911} = \{482\}$,
$\mathcal{T}_{929} = \{468\}$, $\mathcal{T}_{937} = \{665\}$, $\mathcal{T}_{947} = \{435\}$, $\mathcal{T}_{983} = \{456\}$, $\mathcal{T}_{991} = \{16\}$.

Ainsi, on sait que c est égal à 2561 modulo 2^{12}. Avec $M_{\mathfrak{U}} = \prod_{\ell\in\mathfrak{U}} \ell \simeq 1.5 \times 10^{217}$, il n'est pas difficile de trouver à l'aide du théorème chinois que c modulo $M_{\mathfrak{U}}$ est égal à

$$5196350861521230569615182019308761980296483523747909793636843892653731162288473387406548199685790456752967235004941275910801498908400195194700295331211254800056143180853685897049875083716010406996345819386912617948 17.$$

Comme $c \simeq \pm 10^{250}$, il reste donc une trentaine de chiffres décimaux à déterminer pour c. Pour cela, on se tourne maintenant vers l'ensemble $\mathcal{L} \setminus \mathfrak{U}$ des entiers d'Atkin dont les congruences sont,

$$\begin{aligned} \mathcal{T}_{61} &= \pm\{9, 10, 11, 12, 14, 15, 16, 18, 19, 22, 24, 25, 27, 28, 29\}, \\ &\vdots \\ \mathcal{T}_{971} &= \pm\{1, 3, 4, 5, 7, 8, 11, 13, 20, 26, 28, 31, 38, \ldots, 454, 455, 462, 464, 470, 471, 479, 483, 485\}. \end{aligned}$$

Si l'on note C_ℓ, le cardinal des ensembles $\mathcal{T}_\ell$, on peut plus précisément résumer dans le tableau suivant la contribution de chacun de ces entiers.

ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ	ℓ	C_ℓ
61	30	149	40	239	8	313	156	389	48	487	60	617	102	751	92	827	264	887	144
71	12	173	56	241	110	317	52	397	198	499	8	641	212	757	378	829	328	907	452
83	24	181	72	269	72	331	164	409	4	521	56	653	108	761	252	839	48	919	22
97	42	191	32	277	138	349	120	419	24	541	270	659	160	773	84	859	336	941	312
107	4	193	96	281	46	353	58	421	210	557	60	661	330	787	392	863	36	953	312
113	18	197	10	283	140	359	4	443	72	593	180	673	336	809	216	877	438	967	220
137	44	199	20	307	20	367	88	457	228	601	252	683	216	821	272	881	252	971	324
139	12	211	104	311	24	379	8	479	64	607	72	691	344	823	204	883	384		

Comme, $M_{\mathcal{L}\setminus\mathfrak{U}} = \prod_{\ell\in\mathcal{L}\setminus\mathfrak{U}} \ell \simeq 6.2 \times 10^{206}$, à l'évidence, nous avons suffisamment d'entiers d'Atkin pour espérer compléter le calcul. Le problème, l'objet de l'algorithme, est maintenant de distinguer lequel des $C_{\mathcal{L}\setminus\mathfrak{U}} = \prod_{\ell\in\mathcal{L}\setminus\mathfrak{U}} C_\ell \simeq 6.4 \times 10^{150}$ candidats appartient bien à l'intervalle $[-2\sqrt{2^{1663}}, 2\sqrt{2^{1663}}]$.

2.6.2 L'algorithme

Pour faciliter la compréhension, nous décrivons l'algorithme en trois étapes (dans ce qui suit, nous notons M_S le produit des entiers d'un ensemble S).

- L'algorithme naïf. On commence par partitionner l'ensemble

$$\mathfrak{M} = \left\{ \ell \in \mathcal{L} \mid \prod \ell > 4\sqrt{q} \text{ and } C = \prod \#\mathcal{T}_\ell \text{ minimal} \right\},$$

en $\mathfrak{M} = \mathfrak{B} \cup \mathfrak{G}$ tels que $\prod_{\ell\in\mathfrak{B}} C_\ell \simeq \prod_{\ell\in\mathfrak{G}} C_\ell$. Si l'on définit ensuite

$$\mathcal{T}_{\mathfrak{B}/\mathfrak{G}} = \{c \in [-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid \forall \ell \in \mathfrak{B}, c \bmod \ell \in \mathcal{T}_\ell \text{ and } \forall \ell \in \mathfrak{G}, c \bmod \ell = 0\},$$

$$\mathcal{T}_{\mathfrak{G}/\mathfrak{B}} = \{c \in [-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid \forall \ell \in \mathfrak{B}, c \bmod \ell = 0 \text{ and } \forall \ell \in \mathfrak{G}, c \bmod \ell \in \mathcal{T}_\ell\},$$

il n'est pas difficile de voir que

$$\begin{aligned} \mathcal{T}_{\mathfrak{M}} = \{c \in [-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid \forall \ell \in \mathfrak{M}, c \bmod \ell \in \mathcal{T}_\ell\} \\ \subset \{\alpha + \beta + \lambda M_{\mathfrak{M}} \in [-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid (\alpha, \beta) \in \mathcal{T}_{\mathfrak{B}/\mathfrak{G}} \times \mathcal{T}_{\mathfrak{G}/\mathfrak{B}} \text{ et } \lambda \in \{-1, 0, 1\}\}. \end{aligned}$$

Il en découle un algorithme de type "pas de bébés, pas de géants" donné comme suit.

Pas de bébés : pour $\alpha \in \mathcal{T}_{\mathfrak{B}/\mathfrak{G}}$ et pour $\ell \in \mathcal{L} \setminus \mathfrak{M}$, on précalcule les ensembles

$$H_{\alpha,\ell} = \{\theta - \alpha \bmod \ell \mid \theta \in \mathcal{T}_\ell\}.$$

Pas de géants : pour $\lambda \in \{-1, 0, 1\}$ et pour $\beta \in \mathcal{T}_{\mathfrak{G}/\mathfrak{B}}$, si il existe $\alpha \in \mathcal{T}_{\mathfrak{B}/\mathfrak{G}}$ tel que pour tout $\ell \in \mathcal{L} \setminus \mathfrak{M}$, l'entier $\beta + \lambda M_{\mathfrak{M}} \bmod \ell$ est dans $H_{\alpha,\ell}$, alors *c'est gagné*, i. e. $c = \alpha + \beta + \lambda M_{\mathfrak{M}}$ est tel que

$$\forall \ell \in \mathcal{L},\ c \bmod \ell \in \mathcal{T}_\ell.$$

- Une première variante. L'idée de cette première variante est de couper $\mathcal{T}_{\mathfrak{B}/\mathfrak{G}}$ et $\mathcal{T}_{\mathfrak{G}/\mathfrak{B}}$ en de "fines tranches" pour ne considérer que les sommes $\alpha+\beta$ qui sont compatibles avec les congruences modulo un sous-ensemble d'entiers d'Atkin complémentaires $\mathfrak{C}_1 \subset \mathcal{L}\setminus\mathfrak{M}$.

Précisément, soient

$$\mathcal{T}_{\mathfrak{B}/\mathfrak{G}} = \bigcup_{h=0}^{M_{\mathfrak{C}_1}-1} \mathcal{T}^{(h)}_{\mathfrak{B}/\mathfrak{G}} \text{ and } \mathcal{T}_{\mathfrak{G}/\mathfrak{B}} = \bigcup_{h=0}^{M_{\mathfrak{C}_1}-1} \mathcal{T}^{(h)}_{\mathfrak{G}/\mathfrak{B}},$$

où les ensembles $\mathcal{T}^{(h)}_{\mathfrak{B}/\mathfrak{G}}$ et $\mathcal{T}^{(h)}_{\mathfrak{G}/\mathfrak{B}}$ sont donnés pour $h\in\{0,\ldots,M_{\mathfrak{C}_1}-1\}$ par

$$\mathcal{T}^{(h)}_{\mathfrak{B}/\mathfrak{G}} = \{\alpha\in\mathcal{T}_{\mathfrak{B}/\mathfrak{G}} \mid \alpha = h \bmod M_{\mathfrak{C}_1}\},\ \mathcal{T}^{(h)}_{\mathfrak{G}/\mathfrak{B}} = \{\beta\in\mathcal{T}_{\mathfrak{G}/\mathfrak{B}} \mid \beta = h \bmod M_{\mathfrak{C}_1}\}.$$

On note de plus, $\mathcal{T}_{\mathfrak{C}_1} = \{c\in[-M_{\mathfrak{C}_1}/2, M_{\mathfrak{C}_1}/2] \mid \forall \ell\in\mathfrak{C}_1, c \bmod \ell\in\mathcal{T}_\ell\}$. Avec ces notations, l'algorithme devient ce qui suit.

Pour $h\in\{0,\ldots,M_{\mathfrak{C}_1}-1\}$,

Pas de bébés : pour $\Theta\in\mathcal{T}_{\mathfrak{C}_1}$, pour $\alpha\in\mathcal{T}^{((\Theta-h) \bmod M_{\mathfrak{C}_1})}_{\mathfrak{B}/\mathfrak{G}}$ et pour $\ell\in\mathcal{L}\setminus(\mathfrak{M}\cup\mathfrak{C}_1)$, on précalcule les ensembles $H_{\Theta,\alpha,\ell} = \{\theta-\alpha \bmod \ell \mid \theta\in\mathcal{T}_\ell\}$.

Pas de géants : pour $\lambda\in\{-1,0,1\}$ et pour $\beta\in\mathcal{T}^{((h-\lambda M_{\mathfrak{M}}) \bmod M_{\mathfrak{C}_1})}_{\mathfrak{G}/\mathfrak{B}}$, si il existe un entier Θ dans $\mathcal{T}_{\mathfrak{C}_1}$ et un entier α de $\mathcal{T}^{((\Theta-h) \bmod M_{\mathfrak{C}_1})}_{\mathfrak{B}/\mathfrak{G}}$ tel que pour tout entier ℓ de $\mathcal{L}\setminus(\mathfrak{M}\cup\mathfrak{C}_1)$, l'entier $\beta+\lambda M_{\mathfrak{M}} \bmod \ell$ est dans $H_{\Theta,\alpha,\ell}$, alors *on a gagné*, $c=\alpha+\beta+\lambda M_{\mathfrak{M}}$.

- La version finale. L'idée est ici de réduire le stockage nécessaire en découpant $\mathfrak{B}$ et $\mathfrak{G}$ en quatre sous-ensembles $\mathfrak{B}_1$, $\mathfrak{B}_2$, $\mathfrak{G}_1$ et $\mathfrak{G}_2$.

Précisément, soient

$$\mathcal{T}_{\mathfrak{B}_1} = \left\{ c\in[-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid \begin{array}{r} \forall \ell\in\mathfrak{B}_1,\ c \bmod \ell\in\mathcal{T}_\ell \\ \text{et } \forall \ell\in\mathfrak{B}_2\cup\mathfrak{G}_1\cup\mathfrak{G}_2,\ c \bmod \ell = 0 \end{array}\right\}, \mathcal{T}_{\mathfrak{B}_2} = \left\{ c\in[-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid \begin{array}{r} \forall \ell\in\mathfrak{B}_2,\ c \bmod \ell\in\mathcal{T}_\ell \\ \text{et } \forall \ell\in\mathfrak{B}_1\cup\mathfrak{G}_1\cup\mathfrak{G}_2,\ c \bmod \ell = 0 \end{array}\right\},$$
$$\mathcal{T}_{\mathfrak{G}_1} = \left\{ c\in[-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid \begin{array}{r} \forall \ell\in\mathfrak{G}_1,\ c \bmod \ell\in\mathcal{T}_\ell \\ \text{et } \forall \ell\in\mathfrak{B}_1\cup\mathfrak{B}_2\cup\mathfrak{G}_2,\ c \bmod \ell = 0 \end{array}\right\}, \mathcal{T}_{\mathfrak{G}_2} = \left\{ c\in[-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid \begin{array}{r} \forall \ell\in\mathfrak{G}_2,\ c \bmod \ell\in\mathcal{T}_\ell \\ \text{et } \forall \ell\in\mathfrak{B}_1\cup\mathfrak{B}_2\cup\mathfrak{G}_1,\ c \bmod \ell = 0 \end{array}\right\},$$

on a cette fois-ci,

$$\mathcal{T}_{\mathfrak{M}} \subset \{\alpha+\beta+\gamma+\delta+\lambda M_{\mathfrak{M}}\in[-M_{\mathfrak{M}}/2, M_{\mathfrak{M}}/2] \mid (\alpha,\beta,\gamma,\delta)\in\mathcal{T}_{\mathfrak{B}_1}\times\mathcal{T}_{\mathfrak{B}_2}\times\mathcal{T}_{\mathfrak{G}_1}\times\mathcal{T}_{\mathfrak{G}_2} \text{ et } \lambda\in\{-3,\ldots,3\}\}.$$

Si alors, on note $\mathcal{T}_{\mathfrak{B}_2} = \bigcup_{h=0}^{M_{\mathfrak{C}_1}-1} \mathcal{T}^{(h)}_{\mathfrak{B}_2}$ et $\mathcal{T}_{\mathfrak{G}_2} = \bigcup_{h=0}^{M_{\mathfrak{C}_1}-1} \mathcal{T}^{(h)}_{\mathfrak{G}_2}$ où $\mathcal{T}^{(h)}_{\mathfrak{B}_2} = \{\beta\in\mathcal{T}_{\mathfrak{B}_2} \mid \beta = h \bmod M_{\mathfrak{C}_1}\}$, et $\mathcal{T}^{(h)}_{\mathfrak{G}_2} = \{\delta\in\mathcal{T}_{\mathfrak{G}_2} \mid \delta = h \bmod M_{\mathfrak{C}_1}\}$, l'algorithme devient ce qui suit.

Pour $h\in\{0,\ldots,M_{\mathfrak{C}_1}-1\}$,

Pas de bébés : pour $\Theta\in\mathcal{T}_{\mathfrak{C}_1}$, pour $\alpha\in\mathcal{T}_{\mathfrak{B}_1}$, pour $\beta\in\mathcal{T}^{((\Theta-h-\alpha) \bmod M_{\mathfrak{C}_1})}_{\mathfrak{B}_2}$ et pour $\ell\in\mathcal{L}\setminus(\mathfrak{M}\cup\mathfrak{C}_1)$, on précalcule

$$H_{\Theta,\alpha,\beta,\ell} = \{\theta-\alpha-\beta \bmod \ell \mid \theta\in\mathcal{T}_\ell\}.$$

Pas de géants : pour $\lambda\in\{-3,-2,-1,0,1,2,3\}$, pour $\gamma\in\mathcal{T}_{\mathfrak{G}_1}$ et pour $\delta\in\mathcal{T}^{((h-\gamma-\lambda M_{\mathfrak{M}}) \bmod M_{\mathfrak{C}_1})}_{\mathfrak{G}_2}$, si il existe un entier Θ de $\mathcal{T}_{\mathfrak{C}_1}$, un entier $\alpha\in\mathcal{T}_{\mathfrak{B}_1}$ et un entier $\beta\in\mathcal{T}^{((\Theta-h-\alpha) \bmod M_{\mathfrak{C}_1})}_{\mathfrak{B}_2}$ tel que pour tout entier ℓ de $\mathcal{L}\setminus(\mathfrak{M}\cup\mathfrak{C}_1)$, l'entier $\gamma+\delta+\lambda M_{\mathfrak{M}} \bmod \ell$ est dans $H_{\Theta,\alpha,\beta,\ell}$, alors *c'est gagné*, $c=\alpha+\beta+\gamma+\delta+\lambda M_{\mathfrak{M}}$.

Remarque 6. *En pratique, l'algorithme est bien adapté pour les ordinateurs. On stocke les ensembles $H_{\Theta,\alpha,\ell}$ comme un tableau de $\sum_{\ell\in\mathfrak{C}_2}\ell$ chaînes de bits. Les tests de compatibilités de la phase "pas de géants" ne sont plus alors que de simples "ET" logiques.*

2.6.3 Étude de la complexité

L'étude de complexité permet de mesurer le temps et l'espace nécessaire de l'algorithme en fonction de $\log q$ et $C = \#\mathcal{T}_{\mathfrak{M}}$.

En choisissant $\mathfrak{C}_1$ tel que $M_{\mathfrak{C}_1} \simeq O(C^{\frac{3}{8}})$ et en supposant que $C_{\mathfrak{C}_1} \simeq O(C^{\frac{1}{8}})$, alors on peut fixer $\mathfrak{B}_1$, $\mathfrak{B}_2$, $\mathfrak{G}_1$ et $\mathfrak{G}_2$ tel que $C_{\mathfrak{B}_1} \simeq C_{\mathfrak{B}_2} \simeq C_{\mathfrak{G}_1} \simeq C_{\mathfrak{G}_2} \simeq O(C^{\frac{1}{4}})$. Avec ces hypothèses, on peut alors montrer que les complexités de l'algorithme "Chinese & Match" sont les suivantes.

	Précalculs	Pas de bébés	Pas de géants
Espace	$O\left(C^{\frac{1}{4}} \log q \log\log q\right)$	$O\left(C^{\frac{1}{4}} \log^2 q\right)$	$O(1)$
Temps	$O\left(\begin{array}{c} C^{\frac{1}{4}} \log q \log\log q \times \\ \max(\log\log q, \log C) \end{array}\right)$	$O\left(C^{\frac{5}{8}} \max(C^{\frac{1}{8}}, \log^2 q)\right)$	$O(C^{\frac{3}{4}} \log q)$

Au regard des complexités de la méthode "Match et Sort" d'Atkin données ci-dessous, il semble donc que l'algorithme "Chinese & Match" permette au prix d'une faible complexité supplémentaire en temps de diminuer grandement la mémoire nécessaire.

Espace	Temps
$O\left(C^{\frac{1}{2}} \log q\right)$	$O\left(C^{\frac{1}{2}} \log^2 q\right)$

2.6.4 Application à $E(\mathbb{F}_{2^{1663}})$

On choisit $\mathfrak{M} = \mathfrak{U} \cup \{107, 113, 139, 197, 199, 307, 311, 359, 379, 409, 419, 499, 863, 919\}$, alors $M_{\mathfrak{M}} \simeq 5 \times 10^{34} M_{\mathfrak{U}}$, $C_{\mathfrak{M}} \simeq 1.6\ 10^{15}$. On fixe ensuite $\mathfrak{B}_1 = \{199, 307, 919\} \cup \mathfrak{U}$, $\mathfrak{B}_2 = \{113, 197, 419\}$, $\mathfrak{G}_1 = \{107, 359, 379, 409, 863\}$, $\mathfrak{G}_2 = \{139, 311, 499\}$, $\mathfrak{C}_1 = \{239, 839\}$. D'où

$$C_{\mathfrak{B}_1} = 8800,\ C_{\mathfrak{B}_2} = 4320,\ C_{\mathfrak{G}_1} = 18432,\ C_{\mathfrak{G}_2} = 2304, C_{\mathfrak{C}_1} = 384 \text{ et } M_{\mathfrak{C}_1} = 200521.$$

En 1999, après une nuit seulement, sur un réseau de quatre ordinateurs PC cadencés à 300 MHz, on trouva ainsi,

$$\begin{aligned} c = -&380506837724072424056041994887264667366324281439058653016501456962916781408677934 09 \\ &64824975194578376378748556542323117633026752552374659917580100657091283605529024089 5 \\ &38963443344828094088563012564822619018175310540136132830888406802519562014420270438 3. \end{aligned}$$

Remarque 7. *L'un des gros avantages de cet algorithme est qu'il se parallélise aisément. Ici, l'ordinateur numéro i réalisait les itérations $h \in \{50131(i-1), \ldots, 50131i - 1\}$ et une mémoire de 12 mégaoctets seulement fut nécessaire sur chaque machine.*

2.7 Applications

Une implantation de l'algorithme `SEA`, développée sur la base de mes travaux de thèse [104], est maintenant disponible au sein du logiciel de calcul formel MAGMA [19]. Pour cependant illustrer les travaux précédents sous un angle plus cryptographique, nous proposons ici quelques applications issues de ce domaine.

2.7.1 Courbes cryptographiques : la stratégie "Early abort"

La propriété la plus difficile à obtenir lors de la génération au hasard de courbes elliptiques à des fins cryptographiques est d'avoir des nombres de points premiers. Pour en faciliter la recherche, nous avons proposé dans [105] une stratégie efficace. L'algorithme correspondant est le suivant.

1. On choisit aléatoirement une courbe elliptique $E/\mathbb{F}_q$.

2. On calcule $c \bmod \ell$ avec l'algorithme `SEA` pour de petits entiers ℓ et on vérifie que

$$q + 1 - c \bmod \ell \neq 0.$$

   Sinon, cela signifie que la cardinalité de E est divisible par ℓ. Dans ce cas, on retourne à l'étape 1.

3. On termine le calcul de la cardinalité de E avec les algorithmes `SEA` ou Mestre selon la caractéristique de $\mathbb{F}_q$. Si la cardinalité n'a pas un grand diviseur premier, on recommence à l'étape numéro 1.

2.7.2 Améliorations à la "descente de Weil"

La descente de Weil est une méthode qui permet de ramener le logarithme discret sur courbes elliptiques définies sur une extension $\mathbb{F}_{p^n}$ d'un corps fini petit $\mathbb{F}_{p^m}$ à celui du logarithme discret pour une courbe de plus grand genre définie sur $\mathbb{F}_{p^m}$. Elle fut initialement proposée par Frey [49], approfondie par Galbraith et Smart [51], puis rendue efficace par Gaudry, Hess et Smart [56].

Très rapidement, pour le cas $p = 2$, étant donné une courbe $E/\mathbb{F}_{2^n}$, on considère la restriction aux scalaires $W_{E/\mathbb{F}_{2^m}}$ de E sur $\mathbb{F}_{2^m}$ pour m divise n. C'est une variété abélienne de dimension n sur $\mathbb{F}_{2^m}$. La variété est alors intersectée par $n-1$ hyperplans choisis de façon à obtenir une courbe hyperelliptique C définie sur $\mathbb{F}_{2^m}$.

On est alors en mesure de transporter le problème du logarithme discret de E vers C. Si le genre de C est "petit", on est capable de résoudre efficacement le problème du logarithme discret sur E. Pour par exemple $\mathbb{F}_{2^{155}}$, Menezes et Qu ont exhibé de l'ordre de 2^{31} courbes elliptiques pour lesquelles le genre est égal à 11 ou 12.

Or, la descente de Weil est de difficulté variable pour des courbes issues les unes des autres par isogénie. Pour améliorer l'attaque, étant donnée une courbe E sur laquelle on cherche à résoudre le logarithme discret, la stratégie suivante a été avancée par Galbraith, Hess et Smart [50].

1. On recherche parmi les courbes faibles pour la descente de Weil une courbe E' de même cardinalité.
2. On recherche un "chemin" d'isogénies de petit degré qui va de E à E' (c'est la partie difficile).
3. On calcule les isogénies associées au chemin (à l'aide de l'un des algorithmes précédents) et on résout le problème du logarithme discret.

Au final, il s'avère ainsi que sur $\mathbb{F}_{2^{155}}$, ce ne sont pas, à isomorphisme près, 2^{31} courbes qui sont faibles, mais de l'ordre de 2^{100}.

2.7.3 Protections cryptographiques aux attaques physiques

Les attaques physiques des algorithmes et protocoles cryptographiques sont depuis le milieu des années 90 une menace considérée comme sérieuse par la communauté scientifique. Brièvement, on peut distinguer les attaques passives, initialement mises en évidence par Kocher [91] (l'attaquant n'utilise que les informations disponibles à l'extérieur du système comme les temps d'exécution, la puissance électrique consommée, etc.) et les attaques actives, d'abord étudiées par Boneh, DeMillo et Lipton [17] (on perturbe le fonctionnement du système étudié en modifiant son environnement, par exemple l'alimentation électrique, la température, les rayonnements électromagnétiques, etc.)

Bien sûr, la cryptographie à base de courbes elliptiques, en particulier les exponentiations qu'elle implique, n'échappe pas à ces attaques. Les méthodes de protection les plus populaires sont celles proposées par Coron [36] : la randomisation des exposants secrets utilisés dans les algorithmes cryptographiques (en leur additionnant un multiple de la cardinalité de la courbe), l'addition d'un point aléatoire au générateur du groupe et l'utilisation de coordonnées projectives randomisées.

Cette dernière contre-mesure a été mise en défaut par Goubin [59] qui utilise les spécificités de certains points de la courbe utilisée, typiquement les points d'abscisses ou d'ordonnées nulles. Il est cependant possible de contrer les attaques correspondantes, comme proposé par Smart [161], en utilisant dans le calcul cryptographique une courbe isogène aléatoire. Cette mesure nécessite le calcul explicite d'isogénies entre courbes elliptiques, c'est-à-dire la mise en œuvre des algorithmes précédents.

2.7.4 Autres applications de “Chinese & Match”

L’algorithme “Chinese & Match” a une portée qui dépasse le domaine du comptage de points sur courbes elliptiques. Il a notamment été utilisé avec succès

- en 2001 par Jaulmes et Joux et améliorer l’attaque exhaustive d’un protocole cryptographique d’identification [71],
- en 2002 par Chose, Joux et Mitton pour améliorer les attaques par corrélations sur les algorithmes cryptographiques de chiffrement par flot [30].

Cardinalités de courbes hyperelliptiques

— *L'algorithme de Mestre*

Dans [88], Koblitz attire l'attention sur le fait que les Jacobiennes de courbes hyperelliptiques pourraient être une alternative intéressante aux courbes elliptiques comme groupes pour lesquels le problème du logarithme discret est difficile, principalement à cause de l'existence de représentations compactes pour leurs points rationnels et d'algorithmes efficaces pour calculer la loi de groupe [21]. Afin d'éviter des exemples de tels groupes pour lesquels le problème du logarithme discret soit facile, une condition préalable à l'usage de variétés Jacobiennes définies sur des corps finis est le calcul de leur cardinalité. L'histoire des algorithmes pour compter des points sur les Jacobiennes de courbes hyperelliptiques est semblable à celle des courbes elliptiques. Dans [140], Pila décrit une adaptation de l'algorithme de Schoof [154] au cas des courbes hyperelliptiques. De complexité certes polynomiale mais malheureusement de haut degré en la taille du corps, il semble difficile d'utiliser cet algorithme en pratique [52, 53].

Dans le cas des corps finis de petite caractéristique, la situation est toute autre avec l'avènement des méthodes p-adiques. Ces méthodes peuvent être divisées en deux catégories principales.

- Les méthodes cohomologiques qui calculent l'action du Frobenius sur les groupes de cohomologie définis sur un corps de caractéristique zéro. Ces méthodes reposent sur la cohomologie de Monsky-Washnitzer comme décrit dans [84] et [42] ou sur la cohomologie Dwork comme dans [98]. Les deux techniques mènent à des algorithmes d'efficacité raisonnable pour courbes de tout genre.
- Les méthodes de relèvement qui calculent l'action du Frobenius sur le relevé canonique de la Jacobienne. Ceci mène à une méthode très efficace pour les courbes hyperelliptiques définies sur un un corps de caractéristique deux comme décrit dans [125].

Les travaux que l'auteur a mené en collaboration avec Lubicz dans ce domaine ont conduit à un algorithme p-adique efficace, une variante de l'algorithme de Mestre [124]. D'abord spécialisé au cas des courbes elliptiques [108, 62], il est généralisé au cas des courbes hyperelliptiques dans [109]. Plus précisément, nous prouvons le résultat suivant.

Théorème 14. *Soit $\kappa = \mathbb{F}_{2^n}$, un corps fini de caractéristique deux et X, une courbe hyperelliptique définie sur κ dont la Jacobienne est ordinaire et κ-simple, alors, il existe un algorithme de complexité asymptotique en temps égal à $O(n^{2+o(1)})$ et en espace égale à $O(n^2)$ pour calculer le polynôme caractéristique du Frobenius agissant sur la courbe X.*

Les conditions sur la Jacobienne de X dans le théorème 14 sont génériques. Ceci signifie en pratique, que si nous choisissons une courbe "au hasard", l'algorithme retourne presque toujours le bon résultat.

Une fois retracé un historique des travaux apparentés (paragraphe 1), nous abordons ces idées sous un angle mathématique (paragraphe 2) avant de décrire plus en détail les algorithmes associés (paragraphe 3).

1 Travaux apparentés

1.1 Courbes elliptiques

On doit à Satoh [145] d'avoir remarqué que les résultats de Lubin, Serre et Tate [116] conduisent à un algorithme permettant de calculer efficacement le relevé canonique d'une courbe elliptique définie sur un

corps fini de petite caractéristique. Il en déduit un algorithme de complexité $O(n^{3+\varepsilon})$ en temps et $O(n^3)$ en espace pour le calcul du nombre de points. Peu après, Fouquet, Gaudry et Harley [48], et indépendamment Skjernaa [160], explicitent et expérimentent l'algorithme pour les corps finis de caractéristique 2.

L'année d'après, Vercauteren, Preneel et Vandewalle [168] réduisent la complexité en espace à $O(n^2)$. Cependant, à peu près à la même période, Mestre propose un algorithme d'une étonnante simplicité et pourtant de même complexité pour calculer la cardinalité d'une courbe elliptique, la méthode AGM [124].

Viennent alors des améliorations plus algorithmiques. Tout d'abord, Satoh, Skjernaa et Taguchi [147, 146] décrivent une variante des améliorations de [168] dont la complexité est $O(n^{2.5+\varepsilon})$ en temps et $O(n^2)$ en espace. Kim et ses coauteurs expliquent alors que cette complexité très légèrement plus basse, de l'ordre de $O(n^{2.49+\varepsilon})$, lorsque l'on dispose d'une base normale de type gaussienne pour calculer dans une extension non-ramifiée des p-adiques. Gaudry explique ensuite que ces améliorations s'adaptent aussi à l'algorithme de Mestre [54].

Enfin, d'abord Lercier et Lubicz [108], pour les extensions p-adiques avec base normale gaussienne, et ensuite Harley [61] qui généralise, exhibent des algorithmes dont la complexité est $O(n^{2+\varepsilon})$ en temps et $O(n^2)$ en espace.

1.2 Courbes hyperelliptiques

1.2.1 Courbes de genre quelconque

En toute généralité, Lauder et Wan [100, 97] proposent des algorithmes reposant sur la cohomologie de Dwork permettant de compter en temps polynomial le nombre de points rationnels de variétés algébriques générales. Ils en déduisent un algorithme de complexité $O(g^{6+\varepsilon}n^{3+\varepsilon})$ en temps et $O(g^3n^3)$ en espace pour calculer la fonction zêta de courbes d'Artin-Schreier [98].

Kedlaya propose en 2001 un algorithme tirant partie de la cohomologie de Monsky-Washnitzer dont la complexité est $O(g^{5+\varepsilon}n^{3+\varepsilon})$ en temps et toujours $O(g^3n^3)$ en espace. Cet algorithme fut étendu en 2002 par Denef et Vercauteren au cas des courbes d'Artin-Schreier [42] puis aux courbes hyperelliptiques [166] en caractéristique 2. De leur côté, Lauder et Wan atteignent aussi ces complexités [99].

1.2.2 Courbes de petit genre

Pour les courbes de petit genre, Mestre explique dès 2000 que la méthode AGM pour les courbes elliptiques peut être généralisée au cas des courbes de genre 2 [124] avec même complexité ($O(n^{3+\varepsilon})$ en temps et $O(n^2)$ en espace). Il étend en 2002 la méthode aux courbes hyperelliptiques de genre petit en introduisant les fonctions thêta [125]. Cela mène à un algorithme de complexité $O(p^{2g}n^{3+\varepsilon})$ en temps et $O(p^gn^2)$ en espace. Finalement, avec des améliorations d'ordre algorithmique, Lercier et Lubicz expliquent que la complexité peut descendre à $O(p^{2g}n^{2+\varepsilon})$ en temps et $O(p^gn^2)$ en espace [109].

1.3 Autres courbes

On a maintenant bon espoir d'avoir des algorithmes de complexité cubique pour de plus larges familles de courbes. Citons une première extension de Gaudry et Gurel [55] pour le cas des courbes superelliptiques ($O(g^{4+\varepsilon}n^{3+\varepsilon})$ en temps et $O(g^3n^3)$ en espace), puis surtout de Denef et Vercauteren [167] pour le cas plus général des courbes $C_{a,b}$ ($O(g^{5+\varepsilon}n^{3+\varepsilon})$ en temps et $O(g^3n^3)$ en espace).

Enfin notons que Ritzenthaler [143] a généralisé l'approche de Mestre au cas des courbes de genre 3 quelconques, ce qui, couplé avec les améliorations algorithmiques de [109], conduit aussi à $O(p^{2g}n^{2+\varepsilon})$ en temps et $O(p^gn^2)$ en espace.

2 Justifications mathématiques

Soit C une courbe projective algébrique lisse de genre g définie sur un corps $\kappa = \mathbb{F}_q$ avec $q = p^n$. Soit $\overline{\kappa}$ la clôture algébrique de κ. Nous rappelons que le Frobenius σ (resp. le n-ième itéré du Frobenius, F) est l'isogénie associée à l'inclusion entre le corps de fonctions $\kappa(C)$ de C et le sous-corps de fonctions des puissances p-ième (resp. p^n-ième) de $\kappa(C)$ auquel correspond une courbe C' (resp. C). Nous pouvons

définir l'ensemble $C(\kappa)$ des points κ-rationnels de C comme les points de C sur $\overline{\kappa}$ qui sont invariants sous l'action de F.

Alors nous pouvons associer à C sa variété Jacobienne, J, qui est une variété de groupe de dimension g définie sur κ. Pour un nombre entier ℓ premier avec p, nous notons par $J[\ell]$ le sous-groupe des points de ℓ-torsion de J. Il est isomorphe comme groupe à $(\mathbb{Z}/\ell\mathbb{Z})^{2g}$. De plus, si pour $n \in \mathbb{N}$, $[n]$ signifie la multiplication par n dans J, alors, pour $i > j$, nous avons une projection canonique $J[\ell^i] \longrightarrow J[\ell^j]$ donnée par le morphisme $[\ell^{i-j}]$ qui vérifie les relations de compatibilité habituelles, si bien que nous pouvons définir le module de Tate comme la limite projective $T_\ell = \varprojlim J[\ell^i]$. C'est un $\mathbb{Z}_\ell$-module libre de dimension $2g$. Maintenant les éléments du groupe de Galois $\mathrm{Gal}(\overline{\kappa}/\kappa)$ agissent sur J et par conséquent sur $J[\ell]$ d'une façon cohérente avec la projection, si bien que nous avons un morphisme $\rho_\ell : \mathrm{Gal}(\overline{\kappa}/\kappa) \longrightarrow \mathrm{Aut}_{\mathbb{Z}_\ell}(T_\ell)$, appelé la représentation ℓ-adique. Si l'on tensorise de plus T_ℓ par $\mathbb{Q}_\ell$ sur $\mathbb{Z}_\ell$, nous obtenons un morphisme $\rho_\ell : \mathrm{Gal}(\overline{\kappa}/\kappa) \longrightarrow \mathrm{Aut}_{\mathbb{Q}_\ell}(\mathbb{Q}_\ell^{2g})$.

En conséquence, nous pouvons définir le polynôme caractéristique χ_F de F qui est un générateur topologique de $\mathrm{Gal}(\overline{\kappa}/\kappa)$. Rappelons que χ_F est un polynôme unitaire à coefficients entiers et que l'hypothèse de Riemann pour les courbes assure que si nous posons $\chi_F(x) = \prod(x - \lambda_i)$ avec $\lambda_i \in \mathbb{C}$, alors $|\lambda_i| = q^{1/2}$. De plus, il est bien connu que $\chi_F(1)$ est simplement le nombre de points κ-rationnels de $J(C)$. Notre dessein est de donner un algorithme efficace pour calculer χ_F pour les courbes projectives lisses hyperelliptiques de genre g définies sur une extension κ de degré n sur $\mathbb{F}_2$.

2.1 Notations, nombres p-adiques

Nous reprenons rapidement l'approche suivie dans [137]. Les p-adiques ont été inventés au début du vingtième siècle par le mathématicien Kurt Hensel (1861–1941). Le fait est que tout entier $f \in \mathbb{N}$ admet pour tout nombre premier p un développement p-adique de la forme $a_0 + a_1 p + \cdots + a_n p^n$ avec $0 \leqslant a_i < p$. Dès lors que l'on essaie d'écrire un développement de cette forme pour un nombre négatif, on est contraint à un développement infini. L'ensemble des développements de la forme $a_0 + a_1 p + a_2 p^2 + \cdots$ est l'ensemble des entiers p-adiques, il est noté $\mathbb{Z}_p$. Il contient non seulement les entiers, mais aussi les rationnels dont le dénominateur n'est pas divisible par p. Si l'on considère plus largement l'ensemble des développements qui commencent avec une puissance négative de p, i. e. l'ensemble $\mathbb{Q}_p$ des nombres p-adiques, il n'est pas difficile de montrer qu'alors tout rationnel peut être représenté de cette façon.

Montrer que $\mathbb{Z}_p$ muni des lois $+$ et $\times$ est un anneau dont le corps des fractions est $\mathbb{Q}_p$ est techniquement compliqué. Cela est beaucoup plus aisé si l'on définit les entiers p-adiques, étant donné π_i la projection canonique de $\mathbb{Z}/p^{i+1}\mathbb{Z}$ sur $\mathbb{Z}/p^i\mathbb{Z}$, par une suite $x = (x_1, x_2, \ldots, x_i, \ldots)$ avec $x_i \in \mathbb{Z}/p^i\mathbb{Z}$ et $\pi_i(x_{i+1}) = x_i$. Autrement dit, on définit $\mathbb{Z}_p$ comme la limite projective $\varprojlim \mathbb{Z}/p^i\mathbb{Z}$.

Une approche alternative est, tout comme $\mathbb{R}$ peut être construit comme la complétion de $\mathbb{Q}$ en regard de la valeur absolue usuelle $|\ |_\infty$, de considérer que $\mathbb{Q}_p$ est la complétion de $\mathbb{Q}$ avec la valeur absolue p-adique $|\ |_p$ définie pour un nombre rationnel f comme étant égal à $1/p^{v(f)}$ où la valuation canonique v appliquée à f est la plus grande puissance de p divisant f. Ainsi, les nombres p-adiques apparaissent de façon naturelle comme les limites de Cauchy de suites de nombres rationnels. De plus $\mathbb{Q}_p$ est un corps local, son anneau de valuation $\mathbb{Z}_p = \{x \in \mathbb{Q}_p \mid |x|_p \leqslant 1\}$ a un unique idéal maximal, $p\mathbb{Z}_p = \{x \in \mathbb{Q}_p \mid |x|_p < 1\}$ et $\mathbb{Z}_p/p\mathbb{Z}_p \cong \mathbb{F}_p$.

Nous utilisons par la suite la notation K pour désigné l'unique extension non-ramifiée de degré n de $\mathbb{Q}_p$ si $q = p^n$. C'est aussi un corps local d'anneau de valuation $\mathbb{Z}_q = \{x \in K \mid |x|_K \leqslant 1\}$ et d'idéal maximal défini par $p\mathbb{Z}_q = \{x \in K \mid |x|_K < 1\}$ où $|\ |_K$ est l'unique extension de $|\ |_p$ à K. Le corps K est une extension galoisienne de $\mathbb{Q}_p$. Un générateur du groupe de Galois est le Frobenius que nous notons σ. Le Frobenius σ se réduit modulo $p\mathbb{Z}_q$ sur le Frobenius de $\mathbb{F}_q$.

2.2 Fonctions thêta avec caractéristiques rationnelles

Soit V un $\mathbb{C}$-espace vectoriel de dimension g, et puisque $\mathbb{R} \subset \mathbb{C}$, nous pouvons regarder V comme un $\mathbb{R}$-espace vectoriel de dimension $2g$. Soit Λ un réseau de V, c'est-à-dire un sous-groupe discret de V de rang $2g$ sur $\mathbb{Z}$. Nous considérons l'espace quotient $W = V/\Lambda$ et la projection canonique $\pi : V \longrightarrow W$. Il

est bien connu que W hérite alors de la structure de variété analytique de V et que V est le recouvrement universel du tore analytique W.

Afin de caractériser parmi ces tores ceux qui sont algébriques, nous prenons une base e de V et nous identifions V avec $\mathbb{C}^g$ comme $\mathbb{C}$-espace vectoriel par celle-ci. Dans une telle base, Λ peut être représenté par une matrice $g \times 2g$ à coefficients dans $\mathbb{C}$, dont les vecteurs colonnes sont des générateurs de Λ donnés dans la base e. On peut montrer alors qu'un tore est analytiquement isomorphe à une variété analytique projective si et seulement si nous pouvons choisir e telle que, dans une telle base, Λ est donné par une matrice de la forme $(\Delta_g \Omega)$ avec Δ_g une matrice diagonale de dimension g et Ω une matrice symétrique à coefficients complexes telle que Im $\Omega > 0$. Si Δ_g est la matrice identité I_g, alors Λ est dit admettre une polarisation principale, ce que nous considérons toujours dans ce qui suit. Nous notons par $\mathcal{H}_g$ le demi-plan supérieur de Siegel, i. e l'ensemble de telles matrices Ω. Tout point $x \in \mathbb{C}^g$ peut être alors écrit comme $x = \eta I_g + \varepsilon\Omega$, où $\left[\begin{smallmatrix}\eta\\ \varepsilon\end{smallmatrix}\right] \in (\mathbb{R}^g)^2$ est appelé la caractéristique de x. Avec ces conditions W est une variété abélienne.

On peut associer à cette variété abélienne sa fonction thêta de Riemann, vérifiant les équations fonctionnelles suivantes,

$$\theta(-z,\Omega) = \theta(z,\Omega), \quad \theta(z+\alpha,\Omega) = \theta(z,\Omega),$$
$$\theta(z+\Omega\alpha,\Omega) = \exp(-\pi i\, {}^t\alpha\Omega\alpha - 2\pi i^t \alpha z)\theta(z,\Omega),$$

pour $z \in \mathbb{C}^g$ et $\alpha \in \mathbb{Z}^g$. Une telle fonction est donnée par des séries convergentes de fonctions holomorphes,

$$\theta(z,\Omega) = \sum_{N\in\mathbb{Z}^g} \exp(\pi i\, {}^t N\Omega N + 2\pi i\, {}^t Nz).$$

Plus généralement, on peut considérer des fonctions sur V qui réalisent les équations fonctionnelles suivantes. Pour $z \in \mathbb{C}^g$, $\eta, \varepsilon \in \mathbb{Q}^g$, $\alpha \in \mathbb{Z}^g$,

$$\theta\left[\begin{smallmatrix}\eta\\ \varepsilon\end{smallmatrix}\right](z+\alpha,\Omega) = \exp(2\pi i\, {}^t\eta\alpha)\theta\left[\begin{smallmatrix}\eta\\ \varepsilon\end{smallmatrix}\right](z,\Omega),$$
$$\theta\left[\begin{smallmatrix}\eta\\ \varepsilon\end{smallmatrix}\right](z+\Omega\alpha,\Omega) = \exp(-2\pi i\, {}^t\varepsilon\alpha)\exp(-\pi i\, {}^t\alpha\Omega\alpha - 2\pi i\, {}^t\alpha z)\theta\left[\begin{smallmatrix}\eta\\ \varepsilon\end{smallmatrix}\right](z,\Omega).$$

Une telle fonction est appelée une fonction thêta avec caractéristique $[\eta, \varepsilon]$. Elle est donnée en fonction des fonctions thêta de Riemann comme suit.

$$\theta\left[\begin{smallmatrix}\eta\\ \varepsilon\end{smallmatrix}\right](z,\Omega) = \exp(\pi i\, {}^t\eta\Omega\eta + 2\pi i\, {}^t\eta(z+\varepsilon))\theta(z+\Omega\eta+\varepsilon,\Omega).$$

Il existe de nombreuses relations entre ces fonctions thêta. Parmi celles-ci, nous donnons celles qui sont nécessaires pour l'algorithme de Mestre.

2.2.1 Les formules de duplications de Riemann

Les formules de duplications suivantes [47, p. 3],

$$\theta\left[\begin{smallmatrix}\eta\\ \varepsilon\end{smallmatrix}\right](2z_1,2\Omega)\theta\left[\begin{smallmatrix}\eta'\\ \varepsilon\end{smallmatrix}\right](2z_2,2\Omega) = \frac{1}{2^g}\sum_{e\in(\mathbb{Z}/2\mathbb{Z})^g}(-1)^{4\,{}^t\eta e}\theta\left[\begin{smallmatrix}\eta+\eta'\\ \varepsilon+e\end{smallmatrix}\right](z_1+z_2,\Omega)\theta\left[\begin{smallmatrix}\eta-\eta'\\ e\end{smallmatrix}\right](z_1-z_2,\Omega), \qquad (1)$$

pour $z_1, z_2, \in \mathbb{C}^g$ et $\eta, \eta', \varepsilon \in \frac{1}{2}\mathbb{Z}^g$ nous permettent de calculer les valeurs de fonctions thêta lorsque la matrice des périodes est doublée. Notons que ces formules s'appliquent à n'importe quelle variété abélienne.

2.2.2 Les formules de Thomae-Fay

Les formules de Thomae-Fay s'appliquent seulement aux variétés abéliennes qui proviennent de Jacobiennes de courbes hyperelliptiques [1] définies sur $\mathbb{C}$. Soit X une courbe hyperelliptique définie sur $\mathbb{C}$ de genre g donnée dans $\mathbb{P}^2_{\mathbb{C}}$ par une équation de la forme

$$y^2 + h(x)y = u(x), \qquad (2)$$

1. Une généralisation existe pour les courbes de genre 3 suite aux travaux de Ritzenthaler [143].

avec $\deg u(x) = 2g+2$ et $\deg h(x) \leqslant g+1$. C'est une conséquence d'un théorème de Riemann qu'une telle courbe est définie à isomorphisme près par ses points de ramification dans $\mathbb{P}^1(\mathbb{C})$, $B = \{a_1, \ldots, a_{2g+2}\}$. Pour simplifier, nous supposons dans la suite $\deg(h^2 - 4q) = 2g+2$ et que donc $B \subset \mathbb{C} \subset \mathbb{P}^1_{\mathbb{C}}$. Nous savons que $H_1(X, \mathbb{Z})$ est un $\mathbb{Z}$-module libre de rang $2g$. Une base de ce module est formée par des A-cycles et des B-cycles dont les images par la projection $(x, y) \longrightarrow x$ peuvent être dessinées comme sur la figure 1.

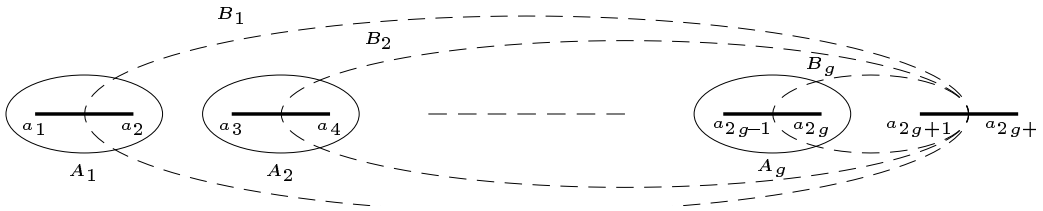


FIGURE 1 – Une base de $H_1(X, \mathbb{Z})$ pour X, une courbe hyperelliptique.

Soient $S = \{a_1, a_3, \ldots, a_{2g+1}\}$, le sous-ensemble des points d'indices impairs de B et U_i, l'ensemble des paires $\{a_{2i-1}, a_{2i}\}$ pour $i = 1, \ldots, g$. Soit $V^g_{1/2}$ l'ensemble des vecteurs de dimension g dont les composantes sont dans l'ensemble $\{0, 1/2\}$ et pour $\varepsilon \in V^g_{1/2}$, nous posons $U_\varepsilon = \cup_j U_j$ avec j parcourant l'ensemble des indices non-nuls dans ε. Avec ces définitions, les formules de Thomae-Fay sont [135, p. 120],

$$\theta \left[\begin{smallmatrix} 0 \\ \varepsilon \end{smallmatrix}\right]^4 (0, \Omega) = \pm\zeta \prod_{i,j \in S \circ U_\varepsilon, i<j} (x_{a_i} - x_{a_j}) \prod_{i,j \notin S \circ U_\varepsilon, i<j} (x_{a_i} - x_{a_j}) \tag{3}$$

pour tout $\varepsilon \in V^g_{1/2}$, ζ une constante qui peut être calculée explicitement (voir par exemple [47, p. 47]), x_{a_i} les abscisses des points a_i et $\circ$, la différence symétrique définie pour deux ensembles T et T' par $T \circ T' = T \cup T' - T \cap T'$.

2.2.3 Formules de transformation

Il existe une action du groupe symplectique $\mathrm{Sp}_{2g}(\mathbb{Z})$ sur $\mathbb{C}^g \times \mathcal{H}_g$. Si $\left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right) \in \mathrm{Sp}_{2g}(\mathbb{Z})$ avec $(\alpha, \beta, \gamma, \delta) \in M_g(\mathbb{Z})^4$, alors cette action est définie par

$$\left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right) (z, \Omega) = ({}^t(\gamma\Omega + \delta)^{-1} z, (\alpha\Omega + \beta)(\gamma\Omega + \delta)^{-1}).$$

Soit Γ_n, le sous-groupe des éléments $M \in \mathrm{Sp}_{2g}(\mathbb{Z})$ tels que $M \equiv I_{2g} \bmod n$. Avec ces notations, nous avons [134, p. 189], pour tout $z \in \mathbb{C}^g$, $(\eta, \varepsilon) \in V^g_{1/2} \times V^g_{1/2}$ et $\left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right) \in \Gamma_2$,

$$\theta \left[\begin{smallmatrix} \eta \\ \varepsilon \end{smallmatrix}\right]^2 ({}^t(\gamma\Omega + \delta)^{-1} z, (\alpha\Omega + \beta)(\gamma\Omega + \delta)^{-1}) = \pm \det(\gamma\Omega + \delta) \theta \left[\begin{smallmatrix} \eta \\ \varepsilon \end{smallmatrix}\right]^2 (z, \Omega). \tag{4}$$

2.3 L'algorithme de Mestre

Soit $q = 2^n$, $\kappa = \mathbb{F}_q$, K l'extension non-ramifiée de degré n de $\mathbb{Q}_2$, nous donnons ici un aperçu général de l'algorithme de Mestre. Nous commençons par un petit résumé des idées de Satoh pour les courbes elliptiques.

2.3.1 La construction de Satoh pour les courbes elliptiques

Soit E une courbe elliptique ordinaire définie sur $\mathbb{F}_q$. Alors, nous pouvons considérer un relevé $\tilde{E}$ de E dans K. C'est une courbe elliptique définie sur K qui se réduit modulo deux sur E. Toutes ces courbes peuvent être définies, par exemple, par une équation de Weierstrass minimale [159, p. 172] définie sur K qui se réduit modulo deux sur l'équation de E. On peut alors considérer une suite de courbes isogènes $\tilde{E}_i$ avec $\tilde{E} = \tilde{E}_0$ et des morphismes $\sigma_i : \tilde{E}_i \longrightarrow \tilde{E}_{i+1}$. En raison d'un théorème de Lubin-Serre-Tate [116], si σ_i se réduit modulo deux sur le Frobenius, alors la suite $\tilde{E}_i$ converge p-adiquement vers le relevé canonique

E^{can} de E dont l'anneau des endomorphismes, $\text{End}(E^{\text{can}})$, est isomorphe à $\text{End}(E)$. Satoh utilise dans son algorithme le dual du Frobenius car ce dernier est plus facile à manipuler dans le cas des courbes non-supersingulières car il est séparable [145]. La valeur propre unité du Frobenius peut alors être obtenue, soit comme premier coefficient de l'action du Frobenius sur le groupe formel, soit au travers de l'injection $\text{End}(E^{\text{can}}) \longrightarrow K$ donnée par l'action d'un endomorphisme sur les différentielles invariantes [159].

2.3.2 Le cas des courbes hyperelliptiques

Soit X_0 une courbe hyperelliptique de genre g définie sur κ et plongée dans $\mathbb{P}^2_\kappa$ par (2). Dans la suite, nous supposons que la variété Jacobienne J_0 de X_0 est ordinaire, ce qui signifie que son 2-rang est égal à g.

Afin de faire le lien avec les fonctions thêta, nous considérons X, un relevé de X_0 sur K donné par n'importe quel relevé de (2). Nous supposons que les points de ramification $B = \{a_1, \ldots, a_{2g+2}\}$ sont rationnels sur K. Leurs abscisses sont données comme les zéros du discriminant $\Delta = h(x)^2 - 4q(x)^2$ et nous pouvons indexer les points de ramification pour que $a_i \equiv a_{i+1} \bmod 2$. Soit J^1_K la variété Jacobienne de X qui est une variété de groupe projective sur K. Pour définir un modèle analytique pour J^1_K, nous fixons un plongement $\iota : K \longrightarrow \mathbb{C}$ tel que nous pouvons considérer X et J^1_K comme des variétés projectives algébriques sur $\mathbb{C}$.

On identifie alors un sous-groupe des points de 2-torsion de J^1_K qui se réduit sur 0 modulo 2, le noyau de l'isogénie Frobenius. En prenant des conventions identiques à celles de la figure 1 pour les A-cycles et B-cycles, nous pouvons définir une matrice des périodes $(I_g \Omega)$ par rapport à une base de $\mathbb{C}^g$ et un tore analytique $J^1_{\text{an}} = \mathbb{C}^g / \Lambda_\Omega$ tel que le sous-groupe engendré par $\frac{1}{2} I_g$ est le noyau du Frobenius.

Maintenant, si J^2_{an} est défini par le réseau $\Lambda_{2\Omega}$, nous pouvons considérer l'isogénie $J^1_{\text{an}} \longrightarrow J^2_{\text{an}}$ définie par l'inclusion de réseau $2\Lambda_\Omega \subset \Lambda_{2\Omega}$. Par le principe GaGa [156], il existe une isogénie $\sigma^1_{\text{alg}} : J^1_K \longrightarrow J^2_K$ qui, par construction, est obtenue en quotientant J^1_K par le sous-ensemble de 2-torsion qui se réduit à zéro modulo 2. Comme ce sous-groupe est défini sur K, σ^1_{alg} est un K-morphisme de K-variétés. En réitérant, on obtient le diagramme commutatif suivant,

$$\begin{array}{ccccccccc}
J^1_{\text{an}} & \xrightarrow{\sigma^1_{\text{an}}} & \cdots & \xrightarrow{\sigma^{m-1}_{\text{an}}} & J^m_{\text{an}} & \xrightarrow{\sigma^m_{\text{an}}} & \cdots & \xrightarrow{\sigma^{m+n-1}_{\text{an}}} & J^{m+n}_{\text{an}} \\
\downarrow & & & & \downarrow & & & & \downarrow \\
J^1_K & \xrightarrow{\sigma^1_{\text{alg}}} & \cdots & \xrightarrow{\sigma^{m-1}_{\text{alg}}} & J^m_K & \xrightarrow{\sigma^m_{\text{alg}}} & \cdots & \xrightarrow{\sigma^{m+n-1}_{\text{alg}}} & J^{m+n}_K
\end{array} .$$

D'après une généralisation d'un théorème de Lubin-Serre-Tate [116] [23, Theorem 4 p. 3], les Jacobiennes $(J^{m+in}_K)_i$ convergent vers J^m_{can}, le relevé canonique de J^m_0 (voir figure 2).

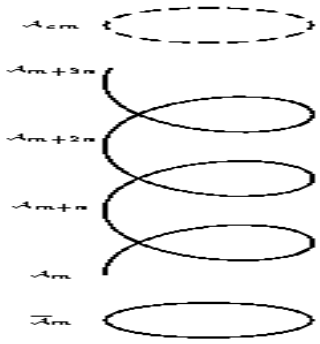


FIGURE 2 – Convergence vers le relevé canonique

2.3.3 Les fonctions thêta revisitées

Comme conséquence de (4) et de la théorie de la multiplication complexe, nous avons

$$\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon\end{smallmatrix}\right](0, 2^n\Omega) = \pm \det(\gamma\Omega + \delta)\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon\end{smallmatrix}\right](0, \Omega).$$

Or $\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon\end{smallmatrix}\right](0, 2^n\Omega)/\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon\end{smallmatrix}\right](0, 2^n\Omega) \equiv 1 \bmod 2^m$. Si nous notons par $\lambda_1, \ldots, \lambda_g$ les valeurs propres unitaires du Frobenius agissant sur X, comme la Jacobienne J^m est ordinaire par hypothèse, le quotient

$$\theta^2 \left[\begin{smallmatrix}0\\ \epsilon\end{smallmatrix}\right](0, 2^n\Omega)/\theta^2 \left[\begin{smallmatrix}0\\ \epsilon\end{smallmatrix}\right](0, \Omega) \tag{5}$$

simplement égal à précision m au produit $\lambda = \pm\lambda_1 \ldots \lambda_g$.

De cette discussion, nous déduisons facilement l'algorithme original de Mestre pour calculer λ à la précision m. Nous choisissons un relevé de X_0 sur K tel que la racine carrée des produits $\prod_{i,j\in S\circ U_\varepsilon, i<j}(x_{a_i} - x_{a_j}) \prod_{i,j\notin S\circ U_\varepsilon, i<j}(x_{a_i} - x_{a_j})$ qui interviennent dans les formules de Thomae-Fay soient définies dans K et $c \in \mathbb{C}$ tel que $c\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon_i\end{smallmatrix}\right](0, \Omega) \in K$. Les formules de Riemann sont homogènes, donc

$$c\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon_i\end{smallmatrix}\right](0, 2^{r+1}\Omega) = \vartheta_i(c\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon_1\end{smallmatrix}\right](0, 2^r\Omega), \ldots, c\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon_{2^g}\end{smallmatrix}\right](0, 2^r\Omega))$$

où ϑ_i est une fraction. Après chaque application des ces formules, les thêta-constantes, $c\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon_i\end{smallmatrix}\right](0, 2^{r+1}\Omega)$ définies dans $\mathbb{C}_2$, sont, relatives à un modèle sur K d'une variété Jacobienne qui est le relevé canonique d'une Jacobienne sur $\mathbb{F}_q$ avec une précision 2-adique, augmentée de un. Une fois que la précision souhaitée est atteinte, n itérations supplémentaires conduisent finalement à λ.

Cet algorithme, de complexité quasi-cubique en n, peut être transformé en un algorithme de complexité quasi-quadratique. On utilise pour cela que les quantités $\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon_i\end{smallmatrix}\right](0, 2^r\Omega)/\theta^2 \left[\begin{smallmatrix}0\\ \varepsilon_{2^g}\end{smallmatrix}\right](0, 2^r\Omega)$ pour $i = 1, \ldots, 2^g - 1$, convergent p-adiquement vers la solution $(\alpha_i)_{i=1\ldots(2g-1)}$ du système d'équations

$$\vartheta_i(\alpha_1, \ldots, \alpha_{2g-1}, 1) = \alpha_i^\sigma, \ i = 1, \ldots, 2^g - 1, \tag{6}$$

où σ est le relevé du Frobenius dans K.

3 Un algorithme de complexité $O(n^{2+\varepsilon})$

La connaissance d'outils algorithmiques efficaces, du moins en caractéristique 2, pour la manipulation des objets mathématiques du paragraphe 1 est récente. Nous en dressons un panorama, d'abord pour les p-adiques, ensuite pour les courbes elliptiques et finalement pour les courbes hyperelliptiques.

3.1 Arithmétique efficace pour les p-adiques

La manipulation des p-adiques sur ordinateur est relativement aisée car l'arithmétique nécessaire sous-jacente est pour l'essentiel une arithmétique d'entiers multi-précision classique. Nous donnons ci-dessous les algorithmes dont on a principalement besoin pour le comptage de points sur des courbes.

3.1.1 Le choix d'une base

Un tout premier besoin est la manipulation explicite des extensions non-ramifiées des p-adiques. Il est nécessaire pour cela de disposer d'une base de K, considéré comme espace vectoriel de dimension n sur le corps $\mathbb{Q}_p$.

- Base polynomiale. Soit $\mathbb{F}_q \cong \mathbb{F}_p[t]/(\overline{F}(t))$, alors K peut-être construit par $K \cong \mathbb{Q}_p[t]/(F(t))$, où $F(t)$ est n'importe quel relevé de $\overline{F}(t)$ dans $\mathbb{Z}_p[t]$. Ce choix conduit à la base polynomiale $\{1, t, \ldots, t^{n-1}\}$. Dans une telle base, il est possible de multiplier efficacement. Typiquement, la multiplication à précision m de deux p-adiques est de complexité $T_{m,n} = O((nm)^\mu)$, où μ dépend de la méthode de multiplication utilisée, par exemple $\mu = 1 + \varepsilon$ avec une méthode de multiplication rapide. Calculer le Frobenius σ d'un élément de K est par contre plus difficile (cf. paragraphe 3.1.3).

- Base normale gaussienne. Pour des extensions de Galois cycliques comme les extensions non-ramifiées K, il existe des éléments α qui conduisent à une base de la forme $\{\alpha, \alpha^\sigma, \ldots, \alpha^{\sigma^{n-1}}\}$.

Au contraire des bases polynomiales, calculer l'image par le Frobenius σ d'un élément de K est immédiat. Les composantes de l'image de x sont simplement une permutation des composantes de x. La multiplication, par contre, est problématique. Une façon astucieuse de lever la difficulté est d'utiliser des bases normales gaussiennes. Par exemple, en relevant une base normale gaussienne définie sur un corps fini de caractéristique deux [120], on a le résultat suivant [85].

Proposition 1. *Soient n, t des entiers strictement positifs tels que $nt+1$ soit un nombre premier impair. Soit γ une racine primitive $(nt+1)$-ième de l'unité dans une extension de $\mathbb{Q}_2$. Si $\mathrm{pgcd}(nt/e, n) = 1$ où e est l'ordre de 2 modulo $nt+1$, alors pour toute racine primitive t-ième de l'unité τ dans $\mathbb{F}_{nt+1}$, l'élément $\alpha = \sum_{i=0}^{t-1} \gamma^{\tau^i}$ génère une base normale sur $\mathbb{Q}_2$ appelée une base normale gaussienne (GNB) de type t. De plus $[\mathbb{Q}_2(\alpha) : \mathbb{Q}_2] = n$.*

Pour une telle base, il n'est pas difficile de voir que la complexité de la multiplication est égale à $T_{m,n} = O((tnm)^\mu)$.

3.1.2 Itérations de Newton

Un algorithme incontournable lorsque l'on travaille avec des p-adiques est l'algorithme fameux de Newton [96, pages 493-494]. Rapidement, ce dernier permet de calculer à précision m la racine d'un polynôme $f(x)$ à coefficients dans K, étant connue cette racine à petite précision. L'algorithme double la précision de cette racine à chaque étape à partir du développement de Taylor,

$$f(x + p^w\delta) = f(x) + p^w\delta\frac{\partial f}{\partial x}(x) + O(p^{2w}).$$

L'algorithme que nous rappelons brièvement ci-dessous est très efficace en pratique. Pour un polynôme avec $O(1)$ termes et de degré $O(1)$, sa complexité en temps est de $O(T_{m,n})$.

Algorithme `Newton`
Algorithm to compute a root of $f(x)$ mod p^m, knowing a solution x_0 modulo p^{2k+1} where $k = v(\partial f/\partial x(x_0))$.
INPUT: $x_0 \in \mathbb{Z}_q/p^{2k+1}\mathbb{Z}_q$, $m \in \mathbb{N}$.
OUTPUT: x a solution of $f(x)$ mod p^m.

1. **if** $m \leqslant 2k+1$ **then return** x_0 ;
2. $w := \lceil \frac{m}{2} \rceil + k$;
3. $x := \texttt{Newton}(x_0, w)$;
4. Lift x to precision m;
5. $V := f(x)$ mod p^m; $\Delta_x := \partial f/\partial x(x)$ mod p^{w-k};
6. **return** $x - V/\Delta_x$;

3.1.3 Relever le Frobenius à précision m

Calculer l'image par le Frobenius d'un p-adique donné par ses composantes dans une base normale est aisé (cf. paragraphe 3.1.1). Sinon, comme suggéré par Harley [61], il est possible de trouver des bases polynomiales définies par un polynôme $F(t)$ telles que $t^\sigma = t^2$. Pour de telles bases, il est alors possible de calculer l'image par σ de tout p-adique x, $x^\sigma = \sum_{i=0}^{n-1} x_i t^{2i}$, avec une complexité de $O(T_{m,n})$.

La technique qu'esquisse Harley (et clairement explicitée dans [167]) est de relever à précision m le polynôme de définition de $\mathbb{F}_q$, $\overline{F}(t)$, à partir de l'équation fonctionnelle,

$$F(t^p) = \prod_{i=0}^{p-1} F(t\zeta^i) \text{ with } \zeta^p = 1.$$

Ceci peut-être fait en temps $O(T_{m,n} \log n)$.

3.2 Courbes elliptiques

3.2.1 Complexité $O(n^{3+\varepsilon})$

Nous donnons ci-dessous, l'algorithme AGM tel que proposé par Mestre pour calculer la cardinalité d'une courbe elliptique sur $\mathbb{F}_{2^n}$. Il n'est pas difficile d'y retrouver les grandes étapes esquissées au paragraphe 2. Sa complexité est celle des étapes 3 et 7, c'est-à-dire $O(n^{3+\varepsilon})$.

Algorithme AGM

Algorithm to compute the trace of an ordinary elliptic curve $E/\mathbb{F}_{2^n} : y^2 + xy = x^3 + \alpha$.

INPUT: $\alpha \in \mathbb{F}_{2^n}^*$.
OUTPUT: The trace c of E.

\\ *Thomae-Fay formulas*
1. $a := 1 + 8\alpha \in \mathbb{Z}_q; b := 1 \in \mathbb{Z}_q$;
\\ *Riemann formulas to reach precision m*
2. for $(i := 1; i < n/2 + O(1); i := i + 1)$ {
3. $a, b := \frac{a+b}{2}, \sqrt{ab}$;
4. }
5. A := a ; B := b ;
\\ *Action of* $\mathrm{Sp}_{2g}(\mathbb{Z})$
6. for $(i := 1; i < n; i := i + 1)$ {
7. $a, b := \frac{a+b}{2}, \sqrt{ab}$;
8. }
\\ *Result*
9. return $\frac{A}{a} \bmod 2^n$ as a signed integer in $[-2\sqrt{2^n}, 2\sqrt{2^n}]$.

3.2.2 Complexité $O(n^{2+\varepsilon})$

En homogénéisant les itérations de Riemann, $a_{i+1} = (a_i + b_i)/2$, $b_{i+1} = \sqrt{a_i b_i}$, par le changement de variable $\tau_i = a_i/b_i$, on obtient $\tau_{i+1} = (2 + \tau_i)/2\sqrt{\tau_i}$. Or, l'équation (6) stipule que $\tau_{i+1} = \tau_i^\sigma$. Par conséquent τ_1 est racine de l'équation

$$4x(x^\sigma)^2 = (1 + x)^2 \ .$$

Cette équation est une équation de la forme $\phi(x, x^\sigma)$ où $\phi(x, y)$ est un polynôme bivarié.

Par ailleurs on peut remarquer que si l'on connaît τ_1 à précision suffisante (de l'ordre de $n/2$) alors la trace de la courbe est égale à la norme

$$c = N_{\mathbb{Z}_{2^n}/\mathbb{Z}_2}\left(2\tau_1/(1 + \tau_1)\right) \ .$$

À l'évidence, nous avons deux problèmes algorithmiques à résoudre : la résolution efficace d'équations de la forme $\phi(x, x^\sigma)$ et le calcul de normes. Nous développons ces deux points ci-dessous.

- Généralisation des itérations de Newton. L'idée principale est de modifier légèrement l'algorithme de Newton afin de conserver une convergence quadratique vers la solution. Soit $\phi \in \mathbb{Z}_p[x, y]$ un polynôme bivarié à coefficients dans $\mathbb{Z}_q$. Soit $x_0 \in \mathbb{Z}_q$ un zéro de l'équation $\phi(x, \Sigma(x)) = 0 \bmod p^w$, $w \in \mathbb{N}$. Nous supposons de plus que

$$v(\frac{\partial \phi}{\partial x}(x_0, \Sigma(x_0))) \geqslant v(\frac{\partial \phi}{\partial y}(x_0, \Sigma(x_0))) \text{ et } v(\phi(x_0, \Sigma(x_0))) > v(\left(\frac{\partial \phi}{\partial y}\right)^2 (x_0, \Sigma(x_0))).$$

Avec pour point de départ le développement de Taylor,

$$\phi(x + p^w\delta, (x + p^w\delta)^\sigma) = \phi(x, x^\sigma) + p^w\delta\frac{\partial \phi}{\partial x}(x, x^\sigma) + p^w\delta^\sigma\frac{\partial \phi}{\partial y}(x, x^\sigma) + O(p^{2w}),$$

on obtient alors l'algorithme suivant où `ArtinSchreierRoot` est une boîte noire qui résout des équations de la forme $x^\sigma = ax + b,\ a$ et b dans $\mathbb{Z}_q$.

Algorithme `NewtonLift`
Algorithm to compute a root of $\phi(x, x^\sigma) \bmod p^m$, knowing a solution x_0 modulo p^{2k+1} where $k = v(\partial\phi/\partial y(x_0, x_0^\sigma))$.
INPUT: $x_0 \in \mathbb{Z}_q/p^{2k+1}\mathbb{Z}_q$, $m \in \mathbb{N}$.
OUTPUT: x a solution of $\phi(x, x^\sigma) \bmod p^m$.

1. if $m \leqslant 2k+1$ **then return** x_0 ;
2. $w := \lceil \frac{m}{2} \rceil + k$;
3. $x :=$ `NewtonLift`(x_0, w);
4. Lift x to $\mathbb{Z}_q/p^m\mathbb{Z}_q$; $y := x^\sigma \bmod p^m$;
5. $\Delta_x := \partial_x\phi(x, y) \bmod p^{w-k}$; $\Delta_y := \partial_y\phi(x, y) \bmod p^{w-k}$;
6. $V := \phi(x, y) \bmod p^m$;
7. $a, b :=$ `ArtinSchreierRoot`$(-V/(p^{w-k}\Delta_y), -\Delta_x/\Delta_y, w-k, n)$;
8. return $x + p^{w-k}(1-a)^{-1}b$;

- Équations d'Artin-Schreier généralisées. Si $\mathbb{F}_q$ est un corps de caractéristique p, une équation d'Artin-Schreier est une équation de la forme $x^p - x + \beta = 0$ avec $\beta \in \mathbb{F}_q$. Une telle équation a une solution si $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(\beta) = 0$. De façon naturelle, nous définissons qu'une équation est une équation d'Artin-Schreier généralisée si elle peut être écrite sous la forme

$$x^\sigma + ax + b = 0, \text{ avec } a, b \in \mathbb{Z}_q. \tag{7}$$

Nous donnons ci-dessous deux algorithmes pour trouver les racines de telles équations, le premier pour les bases normales gaussienne, le second pour les bases polynomiales.

Algorithme `ArtinSchreierRoot (GNB)`
Algorithm to compute a root of $x^\sigma = ax + b$ (when called with $\nu = n$).
INPUT: a and b in $\mathbb{Z}_q/p^m\mathbb{Z}_q$, m and ν in $\mathbb{N}$.
OUTPUT: A and B s.t. $x = Ax^{\sigma^{n-\nu}} + B \bmod p^m$.

1. if $\nu = 1$ **then return** $a^{\sigma^{n-1}}, b^{\sigma^{n-1}} \bmod p^m$;
2. $A, B :=$ `ArtinSchreierRoot`$(a, b, m, \lfloor \frac{\nu}{2} \rfloor)$;
3. $A, B := \begin{cases} AA^{\sigma^{n-\lfloor \frac{\nu}{2} \rfloor}}, \\ AB^{\sigma^{n-\lfloor \frac{\nu}{2} \rfloor}} + B \end{cases} \bmod p^m$;
4. if $\nu \bmod 2$ **then** $A, B := \begin{cases} Aa^{\sigma^{n-\nu}}, \\ Ab^{\sigma^{n-\nu}} + B \end{cases} \bmod p^m$;
5. return A, B;

Algorithme `ArtinSchreierRoot (Polynomial)`
Algorithm to compute a root of $x^\sigma = ax + b$.
INPUT: x, y, V, i, j, a, b such that a and b in $\mathbb{Z}_q/p^m\mathbb{Z}_q$, $v(a) = 1$, $v(b) = 0$, i is the current precision, $j > i$ is the wanted precision,, x a root at precision i, $y = x^\sigma \bmod 2^j$ and $V = y + ax + b$.
OUTPUT: A root x' at precision j and $x'^\sigma \bmod 2^j$.

1. if $j = i + 1$ **then return** $x + 2^i\sqrt{V/2^i}, y + V$;
2. $k := \lfloor \frac{i+j}{2} \rfloor$;
3. $x', y' :=$ `ArtinSchreierRoot`(x, y, V, i, k, a, b) ;
4. $y' := y + (x' - x)^\sigma$;
5. $V+ := (x' - x)a + (y' - y)$;
6. return `ArtinSchreierRoot`(x', y', V, k, j, a, b) ;

On peut voir assez facilement pourquoi l'algorithme `ArtinSchreierRoot` retourne le bon résultat dans le cas des corps avec bases normales gaussiennes. Par une récurrence facile avec point de départ $x^\sigma = ax + b$, on peut écrire que pour tout $k \in \mathbb{N}$, $x^{\sigma^k} \equiv a_k x + b_k \bmod p^m$. On a $x^{\sigma^n} = x$, et donc $(1 - a_n)x = b_n$. Pour calculer a_n et b_n, l'algorithme 3.4 est, à partir de la formule de composition,

$$\forall k, k' \in \mathbb{Z}^2,\ x^{\sigma^{k+k'}} = a_k^{\sigma^{k'}} a_{k'} x + a_k^{\sigma^{k'}} b_{k'} + b_k^{\sigma^{k'}}.$$

une adaptation de la méthode d'exponentiation binaire.

Comprendre l'algorithme dans le cas polynomial est un tout petit peu plus technique, nous référons à [167] pour une explication soignée. Ces deux algorithmes atteignent une complexité de $O(n^{2+\varepsilon})$. Notons cependant que celui pour les bases normales gaussiennes est bien plus efficace en pratique.

- Calcul de normes. Dans tous les cas, la norme $N_{\mathbb{Z}_{2^n}/\mathbb{Z}_2}$ peut être obtenue via un résultant. L'algorithme de Collins en particulier atteint sur nos entrées une complexité de $O(n^{2+\varepsilon})$ [167]. Pour le cas des bases normales gaussiennes, [85] décrit un algorithme de type "divide and conquer", bien plus simple à mettre en œuvre, qui est aussi de complexité $O(n^{2+\varepsilon})$.

3.2.3 Quelques résultats

Nous donnons dans le tableau 1 les temps que nous avons mesurés avec une implantation soignée de l'algorithme en langage `C` avec la librairie `ZEN` [28] compilée au dessus de GMP [57]. Notons qu'indépendamment, Harley [62] a réalisé un calcul pour le corps $\mathbb{F}_{2^{130020}}$.

n	GNB type 1		
	Relevé	Norme	Total
1018	2.5s	1.5s	4s
2052	10s	7s	17s
4098	1mn	45s	1mn 45
8218	6mn 30	4mn 30	11mn
16420	34mn	23mn	57mn
32770	3h 17	2h 18	5h 35
65538	15h 45	13h 20	1j 5
100002	1j 18	1j 16	3j 10

TABLE 1 – Cardinalités de courbes elliptiques sur $\mathbb{F}_{2^n}$ (DEC alpha 731 MHz).

3.3 Courbes hyperelliptiques

Avec le formalisme des fonctions thêta, il est aisé de généraliser aux courbes hyperelliptiques l'algorithme AGM. Par soucis de simplicité, nous décrivons tout d'abord rapidement l'algorithme de complexité $O(n^{3+\varepsilon})$ tel que proposé par Mestre pour les courbes de genre 2 avant d'expliquer plus en détail comment plus généralement, on peut en tirer une variante de complexité $O(n^{2+\varepsilon})$ à genre fixé.

3.3.1 Courbes de genre 2

Pour une courbe de genre 2, une application directe des formules de Thomae-Fay permet de calculer à basse précision les thêta-constantes suivantes.

$$a_0 = \theta \left[\begin{smallmatrix} 0 \\ (0,0) \end{smallmatrix} \right] (0,\Omega),\ b_0 = \theta \left[\begin{smallmatrix} 0 \\ (0,1/2) \end{smallmatrix} \right] (0,\Omega),$$
$$c_0 = \theta \left[\begin{smallmatrix} 0 \\ (1/2,0) \end{smallmatrix} \right] (0,\Omega),\ d_0 = \theta \left[\begin{smallmatrix} 0 \\ (1/2,1/2) \end{smallmatrix} \right] (0,\Omega).$$

Les formules de Riemann, plus connues ici sous le nom de moyenne de Borchardt, sont alors données par,

$$\begin{cases} a_{i+1} &= (a_i + b_i + c_i + d_i)/4, \\ b_{i+1} &= (\sqrt{a_i b_i} + \sqrt{c_i d_i})/2, \\ c_{i+1} &= (\sqrt{a_i c_i} + \sqrt{b_i d_i})/2, \\ d_{i+1} &= (\sqrt{a_i d_i} + \sqrt{b_i c_i})/2. \end{cases}$$

Il suffit alors d'itérer ces formules avec valeurs initiales les thêta-constantes pour que l'on obtienne des solutions A, B, C et D à précision voulue, ici de l'ordre de $3n/2$.

Après n itérations supplémentaires, on obtient $A/a = B/b = C/c = D/d$ qui se trouve être égal au produit des valeurs propres inversibles modulo 2 du Frobenius, produit à partir duquel il est possible de retrouver la fonction zêta de la courbe.

3.3.2 Complexité $O(n^{2+\epsilon})$

Afin de calculer le polynôme caractéristique χ_F du Frobenius F défini sur une courbe hyperelliptique ordinaire de $\mathbb{F}_{2^n}$, l'algorithme que nous considérons dans cette partie est composé de quatre phases principales qui, sont résumées comme suit.

Initialisations : Étant donnée une courbe hyperelliptique X définie explicitement sur un corps fini $\mathbb{F}_{2^n}$, cette phase calcule à petite précision les valeurs prises par 2^g thêta-constantes $\theta_i = c\theta^2 \left[\begin{smallmatrix} 0 \\ \varepsilon_i \end{smallmatrix}\right](0, \Omega)$, $i = 1, \ldots, 2^g$. Ces thêta-constantes sont les valeurs prises à $z = 0$ par les fonctions thêta avec 2^g caractéristiques attachées au réseau de X (cf. paragraphe 2.2).

Relèvements : En utilisant les formules de duplication de Riemann données dans le paragraphe 2.3, on est amené à résoudre un système d'équations multivariées $G(x) = x^\sigma$ dans $\mathbb{Z}_q$ à précision m assez grand. La solution est un vecteur à $2^g - 1$ composantes. Chaque composante d'une telle solution est le quotient d'une constante thêta divisée par une autre constante thêta que l'on fixe. Ceci peut être fait, à partir des thêta-constantes connues à petite précision grâce à un algorithme de relèvement.

Norme : Calculer la norme $\mathrm{N}_{\mathbb{Z}_q/\mathbb{Z}_2}$ d'un élément de $\mathbb{Z}_q$ dérivé des $2^g - 1$ quotients de relevés de thêta-constantes connues à précision m conduit au produit $\lambda_1 \cdots \lambda_g$ de g valeurs propres (inversibles modulo 2) de F.

LLL : Retrouver le polynôme caractéristique χ_F du Frobenius de X peut être fait comme décrit par Mestre. Au début, nous construisons un polynôme symétrique de degré 2^{g-1} dont la racine est $\eta = \lambda + 2^{gn}/\lambda$. Puis, nous calculons les racines de ce polynôme sur $\mathbb{C}$ et les recombinons afin de trouver $\chi_F(\pm x)$.

- Initialisations. Nous supposons dans la suite qu'une courbe hyperelliptique ordinaire X de genre g est définie sur $\mathbb{F}_{2^n}$ par la donnée d'un modèle affine de la forme $y^2 + h(x)y = q(x)h(x)$ où $h(x)$ et $q(x)$ sont des polynômes de $\mathbb{F}_{2^n}$ de degré $g+1$ tels que $h(x)$ a $g+1$ racines de multiplicité exactement 1 sur $\mathbb{F}_{2^n}$. Afin d'obtenir les thêta-constantes à petite précision, nous relevons d'abord dans $\mathbb{Z}_{2^n}$ le modèle affine de X comme $(2y + h(x))^2 = h(x)(h(x) + 2^2 q(x))$. En utilisant le lemme de Hensel, il n'est pas difficile de voir que $h(x)$ et $h(x) + 4q(x)$ se décomposent complètement sur $\mathbb{Z}_{2^n}$, cela conduit à une équation de la forme

$$y^2 = \prod_{i=0}^{2g+1} (x - a_i) \text{ tel que } a_i \in \mathbb{Z}_{2^n}^2 \text{ et } a_{2i} \equiv a_{2i+1} \bmod 2^2.$$

Alors, les formules de Thomae-Fay (cf. (3)) permettent de calculer 2^g thêta-constantes $\theta_0, \ldots, \theta_{2^g-1}$ à petite précision via

$$\theta_e = \sqrt{\prod_{0 \leqslant i < j \leqslant g} (a_{2i+e_i} - a_{2j+e_j})(a_{2i+1-e_i} - a_{2j+1-e_j})},$$

où $e_0 = 0$ et où e est écrit en base deux comme $e_g 2^{g-1} + \cdots + e_1$. La racine carrée est choisie telle que $\theta_e \equiv 1 \bmod 2^2$.

- Relever les courbes. Soit $R(t_1) = \frac{2\sqrt{t_1}}{1+t_1}$ pour les courbes de genre $g = 1$ et

$$R(t_1, \cdots, t_{2^g-1}) = \frac{2\sqrt{t_1} + 2\sqrt{t_2}\sqrt{t_3} + \cdots + 2\sqrt{t_{2^g-2}}\sqrt{t_{2^g-1}}}{1 + t_1 + \cdots + t_{2^g-1}} \text{ sinon.}$$

Les formules de duplication de Riemann donnent alors

$$\forall e \in \{1, \ldots, 2^g - 1\},\ \tau_e^\sigma = R(\tau_e, \tau_{i_2}, \tau_{i_3}, \ldots, \tau_{i_{2^g-2}}, \tau_{i_{2^g-1}})$$

où, pour tout e, les indices $i_2, \ldots, i_{2^g-1}$ sont tels que

$$\{\{0, e\}, \{i_2, i_3\}, \ldots, \{i_{2^g-2}, i_{2^g-1}\}\} = \{\{j, j \oplus e\} \mid j \in \{1, \ldots, 2^g - 1\}\}$$

($\oplus$ signifie le ou exclusif de deux entiers) et $\tau_e = \theta_e/\theta_0 \bmod 2^4$.

Il est possible de généraliser l'algorithme 3.3, `NewtonLift`, pour trouver en $O(n^{2+\varepsilon})$ la solution d'un tel système multivarié d'équations. Nous référons à [109] pour les détails.

- Norme. L'équation (5) donne dans notre cas,

$$\lambda \equiv \mathrm{N}_{\mathbb{Z}_{2^n}/\mathbb{Z}_2}\left(\frac{2^g}{1+\tau_1+\cdots+\tau_{2^g-1}}\right) \bmod 2^m,$$

et on peut d'utiliser des algorithmes identiques à ceux utilisés pour les courbes elliptiques.

- Réduction de réseaux La fin de notre variante est identique à l'algorithme original de Mestre [125]. D'abord, en utilisant l'algorithme de réduction de réseau LLL [101], on obtient un polynôme symétrique $P_{sym}(x)$ dont les racines sont de la forme $x + q/x$ où x est le produit de g termes qui appartiennent à $\{\lambda_1, q/\lambda_1\}, \ldots, \{\lambda_g, q/\lambda_g\}$. Alors nous calculons ses racines sur $\mathbb{C}$ et obtenons à partir d'elles $\chi_F(\pm x)$. Une dernière vérification sur la courbe nous permet d'obtenir χ_F.

Le réseau que nous construisons à partir de $\eta = \pi + 2^{gn}/\pi$ pour retrouver le polynôme $P_{sym}(x)$ est donné par

$$\begin{bmatrix} \Upsilon \times M_1 & \Upsilon \times M_2 & \cdots & \Upsilon \times M_{2^{g-1}+1} & \Upsilon \times 2^m \\ 0 & 0 & \cdots & 2^{\lfloor n \times S_{2^{g-1}+1} \rfloor} & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \\ 0 & 2^{\lfloor n \times S_2 \rfloor} & \cdots & 0 & 0 \\ 2^{\lfloor n \times S_1 \rfloor} & 0 & \cdots & 0 & 0 \end{bmatrix},$$

où

$$M_i = 2^{(2^{g-1}-1-i)n}\eta^i \bmod 2^m \text{ pour } i \in \{0, \ldots, 2^{g-1}-1\},\ M_{2^{g-1}} = \eta^{2^{g-1}} \bmod 2^m,\ M_{2^{g-1}+1} = 2^m,$$

et

$$S_i = \frac{(i-1)(g-2)}{2} \text{ pour } i \in \{1, \ldots, 2^{g-1}\},\ S_{2^{g-1}+1} = \frac{2^{g-1}(g-2)}{2} + 1$$

(Υ est une constante arbitrairement grande).

Précision nécessaire. On peut montrer à partir d'estimations asymptotiques connues [101, 34] que dans le cas présent LLL retournera le polynôme P_{sym} si la précision m à laquelle l'on connaît η satisfait

$$m > \frac{2^{2g}(g-2) + 2^{g+1}(g+2)}{16}\, n\ .$$

On donne des valeurs numériques dans le tableau 2.

g	1	2	3	4	5	6	7	8	9	10
m	$n/2$	$2n$	$9n$	$44n$	$220n$	$1088n$	$5264n$	$24896n$	$115392n$	$525824n$

TABLE 2 – Valeurs asymptotiques de la précision m en fonction de g et n.

3.3.3 Un exemple en genre 4

Afin d'illustrer l'algorithme, nous nous proposons de calculer le polynôme caractéristique du Frobenius sur la courbe hyperelliptique de genre 4 définie sur $\mathbb{F}_{2^4} \simeq \mathbb{F}_2[t]/(t^4+t^3+t^2+t+1)$ dont un modèle affine est $y^2 + h(x)y + q(x)h(x) = 0$ où

$$\begin{aligned} h(x) &= (x + t^3 + t^2 + t + 1)(x + t^2)(x + t^3 + 1)(x + t + 1)(x + t^3 + t^2 + 1), \\ q(x) &= x^5 + \left(t^3 + t^2 + t + 1\right) x^4 + x^3 + t^3 x^2 + \left(t^3 + t + 1\right) x. \end{aligned}$$

Les abscisses des 10 points de 2-torsion d'un relevé de la courbe dans une extension non-ramifiée des 2-adiques définie par $t^4+t^3+t^2+t+1$ sont, à précision 6,

$$a = [-t-1,\ 4\,t^3-24\,t^2-13\,t+15,\ -t^2,\ -28\,t^3+11\,t^2-20\,t-28,\ -13\,t^3+24\,t^2+4\,t-1,\\ -t^3-1,\ 3\,t^3-17\,t^2-8\,t-9,\ -t^3-t^2-1,\ -9\,t^3+27\,t^2+15\,t+23,\ -t^3-t^2-t-1].$$

Les formules de Thomae-Fay conduisent à 15 constantes $\tau_e = \theta_e/\theta_0$ qui, à précision 7, sont égales à

$$\tau = [-32\,t^3+16\,t^2+8\,t-55, -56\,t^3-40\,t^2-32\,t+49, -8\,t^3+24\,t^2-40\,t+9,\\ -48\,t^3+48\,t^2-48\,t+9, 48\,t^3+64\,t^2-40\,t+1, 24\,t^3+24\,t^2+64\,t-55, -56\,t^3-40\,t^2+56\,t-47,\\ -32\,t^3-24\,t^2+56\,t-47, 64\,t^3+24\,t^2-48\,t+57, -40\,t^3-32\,t^2-8\,t-15, 8\,t^3+64\,t^2-23,\\ 48\,t^3-24\,t^2+8\,t+41, 16\,t^3+24\,t^2+32\,t-63, 40\,t^3-16\,t^2-40\,t-39, -40\,t^3-48\,t^2-32\,t+1].$$

Un appel à `NewtonLift` relève τ à précision 345. Ce qui donne en notation hexadécimale,

$$\tau = [-\texttt{2AF4761F43EBADC244C1BC1D33E90C24141C48F828C8E05A9E7ADB00EC35D6F88BD03D0C05C445A4BFF230}\,t^3\\ +\texttt{1F66D441F994F38896AF5CB90E34007AD48632BBBC09695F6E2E5A4ED676ED6752EBE9B9239EACB570F370}\,t^2\\ +\texttt{1F173EE17446547549FBE4BE2CA778C1AA31398B6AB73966621BF4D4A63B45131165F0E1847B040F40E648}\,t\\ +\texttt{487AD2E1552D785AC648ED52E76D6195E111BCB022D02B334512B58E067205652901ADD8E97630C49196A9},$$

$$\vdots$$

],

et

$$\lambda = \texttt{18184F8253A78523EE4F72D801B910F4A83B8B844AAA42D2CC911C89846B4B24D5B0DB3F56FA9354B37C56D} \bmod 2^{345}.$$

Après la réduction de réseau, on trouve

$$P_{sym}(x) = x^8+467\,x^7-2^4\cdot 25988\,x^6-2^8\cdot 837798\,x^5+2^{12}\cdot 9084572\,x^4+\\ 2^{16}\cdot 417375179\,x^3+2^{20}\cdot 1472562912\,x^2-2^{24}\cdot 37930023936\,x-2^{28}\cdot 253989847040,$$

et finalement

$$\chi_F(x) = x^8-2\,x^7+12\,x^6-62\,x^5+339\,x^4-2^4\cdot 62\,x^3+2^8\cdot 12\,x^2-2^{12}\cdot 2\,x+2^{16}.$$

3.3.4 Quelques résultats

Avec des libraires arithmétiques analogues à celles utilisées pour les courbes elliptiques, nous avons programmé en `C` cet algorithme pour les courbes de genre 2 et 3. Nous donnons les résultats dans le tableau 3.

n	GNB type 1 (genre 2)		
	Lift	Norme	Total
1018	2mn	5s	2mn 5
2052	8mn 30	25s	8mn 55
4098	50mn 5s	2mn 15	52mn 20
8218	4h 52mn	13mn	5h 5
16420	1j 5	1h	1j 6
32770	7j 22	6h	8j 4

n	GNB type 1 (genre 3)		
	Lift	Norme	Total
1018	8h 30	1mn	8h 31
2052	1j 3	5mn	1j 3
4098	6j 8h	25mn	6j 8h

TABLE 3 – Fonctions zêta de courbes hyperelliptiques de genre 2 et 3 (DEC alpha 731 MHz)

Bibliographie

[1] L. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. *IEEE*, pages 55–60, 1979. Twentieth annual symposium on foundations of computer science.

[2] G.B. Agnew, R.C. Mullin, and S.A. Vanstone. A fast elliptic curve cryptosystem. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology — EUROCRYPT '89*, pages 706–708, Berlin, 1989. Springer-Verlag. Lecture Notes in Computer Science Volume 434.

[3] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The design and analysis of computer algorithms*. Reading. Addison–Wesley, 1974.

[4] S. Alt and R. Lercier. How to Make a Traceable Block Cipher Stealthy. Preprint, May 2004.

[5] A.O.L. Atkin. The number of points on an elliptic curve modulo a prime, 1988. Email sur la mailing liste NMBRTHRY.

[6] A.O.L. Atkin. The number of points on an elliptic curve modulo a prime, 1991. Email sur la mailing liste NMBRTHRY.

[7] A.O.L. Atkin and F. Morain. Elliptic curves and primality proving. *Mathematics of Computations*, 61 :29–68, 1993.

[8] P. Barreto. Weil and Tate pairings. *http ://planeta.terra.com.br/informatica/paulobarreto/pblounge.html*, 2004.

[9] P. Barrett. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In A.M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, pages 311–326, Berlin, 1986. Springer-Verlag. Lecture Notes in Computer Science Volume 263.

[10] M. Bellare, J. Killian, and P. Rogaway. The Security of the Cipher Block Chaining Message Authentication Code. In *Advances in Cryptology — CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 341–358. Springer, 1994.

[11] M. Bellare and T. Kohno. A Theoretical Treatment of Related-Key Attacks : RKA–PRPs, RKA–PRFs, and Applications. In E. Biham, editor, *Advances in Cryptology — Proceedings of EUROCRYPT 2003*, Lecture Notes in Computer Science. Springer, 2003.

[12] A. Bender and G. Castagnoli. On the implementation of elliptic curve cryptosystems. In G. Brassard, editor, *Advances in Cryptology — CRYPTO '89*, pages 186–193, Berlin, 1989. Springer-Verlag. Lecture Notes in Computer Science Volume 435.

[13] D.J. Bernstein. Floating-Point Arithmetic and Message Authentication, March 2000. Submitted for publication.

[14] D.J. Bernstein. Circuits for integer factorization. A proposal, November 2001.

[15] D.J. Bernstein. Multidigit multiplication for mathematicians. *Advances in Applied Mathematics*, 2002. À paraître.

[16] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC : Fast and Secure Message Authentication. In *Advances in Cryptology – CRYPTO '99*. Springer-Verlag, August 1999.

[17] D. Boneh, R.A. DeMillo, and R.J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, pages 37–51, Berlin, 1997. Springer-Verlag. Lecture Notes in Computer Science Volume 1233.

[18] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In *Crypto '2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, 2001.

[19] W. Bosma, J. Cannon, and C. Playoust. The Magma Algebra System I : The User Language. *J. Symbolic Comp.*, 24(3) :235–265, 1997.

[20] D.-R.-L. Brown. Generic groups, collision resistance, and ECDSA. Available at *http ://eprint.iacr.org/2002/026/*, 2002.

[21] D.G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48(177) :95–101, 1987.

[22] D.G. Cantor. On arithmetical algorithms over finite fields. *Journal of Combinatorial Theory*, 50(285–300), 1989. Series A.

[23] R. Carls. Generalized AGM sequences and approximation of canonical lifts. Available at *http ://www.math.leidenuniv.nl/˜carls*, April 2003.

[24] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of computer and system sciences*, 18 :143–154, 1979.

[25] J.L. Carter and M.N. Wegman. New hash functions and their use in authentication and set equality. *Journal of computer and system sciences*, 22(18) :265–279, 1981.

[26] F. Chabaud and J.-M. Couveignes, editors. *Bulletin de la cellule de prospective en Codage, Cryptographie et Sécurité des Réseaux*, volume II, chapter "Courbes elliptiques et cryptographie". CELAR, 1997.

[27] F. Chabaud and J.-M. Couveignes, editors. *Bulletin de la cellule de prospective en Codage, Cryptographie et Sécurité des Réseaux*, volume III, chapter "Le logiciel de calcul formel MAGMA". CELAR, 1998.

[28] F. Chabaud and R. Lercier. *ZEN, User Manual.* Available at *http ://sourceforge.net/projects/zenfact/*.

[29] J. Chao, K. Tanada, and S. Tsujii. Design of elliptic curves with controllable lower boundary of extension degree for reduction attacks. In Y. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pages 50–55, Berlin, 1994. Springer-Verlag. Lecture Notes in Computer Science Volume 839.

[30] P. Chose, A. Joux, and M. Mitton. Fast Correlation Attacks : An Algorithmic Point of View. In L. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 209–221, Berlin, 2002. Springer.

[31] D.V Chudnovsky and G.V Chudnovsky. Algebraic complexities and algebraic curves over finite fields. *Journal of complexity*, 4 :285–316, 1988.

[32] H. Cohen and G. Frey, editors. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, chapter "Point Counting on Elliptic and Hyperelliptic Curves", R. Lercier, D. Lubicz and F. Vercauteren. Chapman & Hall/CRC, 2005. En préparation.

[33] S.A. Cook. *On the minimum computation time of functions.* PhD thesis, Harvard University, Cambridge, Massachussets, 1966.

[34] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4) :233–260, 1997.

[35] D. Coppersmith, A. Odlyzko, and R. Schroppel. Discrete logarithms in $GF(p)$. *Algorithmica*, 1 :1–15, 1986.

[36] J.-S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In *CHES'99*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302, Berlin, 1999. Springer-Verlag.

[37] J.-M. Couveignes. *Quelques calculs en théorie des nombres.* thèse, Université de Bordeaux I, July 1994.

[38] J.-M. Couveignes. Computing l-isogenies with the p-torsion. In H. Cohen, editor, *ANTS-II*, volume 1122 of *Lecture Notes in Computer Science*, pages 59–65. Springer-Verlag, 1996.

[39] J.-M. Couveignes. Isomorphisms between Artin-Schreier towers. *Mathematics of Computation*, 69(232) :1625–1631, 2000.

[40] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. Available at *http ://shoup.net/*, 2002.

[41] N. Demytko. A new elliptic curve based analogue of RSA. In T. Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, pages 40–49, Berlin, 1993. Springer-Verlag. Lecture Notes in Computer Science Volume 765.

[42] J. Denef and F. Vercauteren. An extension of Kedlaya's algorithm to Artin-Schreier curves in characteristic 2. In D.R. Kohel C. Fieker, editor, *Algorithmic Number Theory, 5th International Symposium, ANTS-V*, pages 369–384. Springer, July 2002.

[43] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, 22(6) :644–654, 1976.

[44] D. Eisenbud. *Commutative Algebra with a View Toward Algebraic Geometry.* Number 150 in Graduate Texts in Mathematics. Springer, 1995.

[45] N.D. Elkies. Explicit isogenies. Preprint, 1991.

[46] N.D. ELkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational Perspectives On Number Theory*, 1997. To appear. In honor of A. O. L. Atkin. Held September, 1995 in Chicago.

[47] J.D. Fay. *Theta functions on Riemann surfaces.* Springer-Verlag, Berlin, 1973. Lecture Notes in Mathematics, Vol. 352.

[48] M. Fouquet, P. Gaudry, and R. Harley. An extension of Satoh's algorithm and its implementation. *Journal of Ramanujan Mathematical Society*, 15 :281–318, 2000.

[49] G. Frey. How to disguise an elliptic curve. Talk at ECC'98, 1998. Waterloo.

[50] S.D. Galbraith, F. Hess, and N.P. Smart. Extending the GHS Weil Descent Attack. In L. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 29–44, Berlin, 2002. Springer.

[51] S.D. Galbraith and N.P. Smart. A Cryptographic application of Weil descent. In *Codes and Cryptography*, volume 1746 of *Lecture Notes in Computer Science*, pages 191–200, Berlin, 1999. Springer-Verlag.

[52] P. Gaudry. *Algorithmique des courbes hyperelliptiques et applications à la cryptologie.* PhD thesis, École Polytechnique, 2000.

[53] P. Gaudry. Cardinality of a genus 2 hyperelliptic curve over $\mathrm{GF}(5 \cdot 10^{24} + 41)$. Email sur la mailing liste NMBRTHRY, September 2002.

[54] P. Gaudry. A comparison and a combination of SST and AGM algorithms for counting points of elliptic curves in characteristic 2. In *Advances in Cryptology — ASIACRYPT 2002*, Lecture Notes in Comput. Sci. Springer, Berlin, December 2002.

[55] P. Gaudry and N. Gurel. An extension of Kedlaya's point-counting algorithm to superelliptic curves. In *Advances in Cryptology — ASIACRYPT 2001*, Lecture Notes in Computer Science, Berlin, 2001. Springer.

[56] P. Gaudry, F. Hesse, and N.P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *Journal of Cryptology*, 15 :19–46, 2002.

[57] Free Software Foundation GNU. GMP library. Available at *http ://www.swox.com/gmp/*, 2002.

[58] D. Gordon. Discrete logarithms in $GF(p)$ using the number field sieve. *SIAM J. Discrete Math*, 6 :124–138, 1993.

[59] L. Goubin. A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. In *PKC'2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 199–211, Berlin, 2003. Springer-Verlag.

[60] H. Gunji. The Hasse invariant and p-division points of an elliptic curve. *Arch. Math.*, 27(2), 1976.

[61] R. Harley. *Algorithmes avancés pour l'arithmétique des courbes.* PhD thesis, Université Paris 7, 2002.

[62] R. Harley. Asymptotically optimal p-adic point-counting. Email sur la mailing liste NMBRTHRY, December 2002.

[63] R. Harley. Elliptic curve point counting : 32003 bits. Email sur la mailing liste NMBRTHRY, August 2002.

[64] G. Harper, A. Menezes, and S.A. Vanstone. Public-key cryptosystems with very small key lengths. In R.A. Rueppel, editor, *Advances in Cryptology — EUROCRYPT '92*, pages 163–173, Berlin, 1992. Springer-Verlag. Lecture Notes in Computer Science Volume 658.

[65] R. Hartshorne. *Algebraic Geometry.* Graduate Texts in Mathematics. Springer, 1977.

[66] J. Hastad. On using RSA with low exponent in a public key network. In H.C. Williams, editor, *Advances in Cryptology — CRYPTO '85*, pages 403–408, Berlin, 1986. Springer-Verlag. Lecture Notes in Computer Science Volume 218.

[67] E. Jaulmes. *Analyse de sécurité de schémas cryptographiques.* PhD thesis, École Polytechnique, June 2003.

[68] E. Jaulmes, A. Joux, and F. Valette. On the Security of Randomized CBC-MAC beyond the Birthday Paradox Limit. A New Construction. In V. Rijmen J. Daemen, editor, *Fast Software Encryption 2002*, volume 2365 of *Lecture Notes in Computer Science*, pages 237–251. Springer Verlag, February 2002.

[69] E. Jaulmes and R. Lercier. FRMAC, a Fast Randomized Message Authentication Code. Submitted for publication at FSE'2005, September 2004.

[70] A. Joux. A one round protocol for tripartite diffie-hellman. In *Fourth Algorithmic Number Theory Symposium*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394, 2000.

[71] A. Joux and E. Jaulmes. Cryptanalysis of PKP : A new Approach. In K. Kim, editor, *PKC'2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 165–172, Berlin, 2001. Springer-Verlag.

[72] A. Joux and R. Lercier. Discrete logarithms in GF(p). Email sur la mailing liste NMBRTHRY, 1998.

[73] A. Joux and R. Lercier. State-of-the-art in implementing algorithms for the (ordinary) discrete logarithm problem. Talk at the Elliptic Curve and Cryptography conference, November 1999.

[74] A. Joux and R. Lercier. Calcul de logarithmes discret dans $GF(2^{521})$. Email sur la mailing liste NMBRTHRY, September 2001.

[75] A. Joux and R. Lercier. "Chinese & match", an alternative to Atkin's "match and sort" method used in the SEA algorithm. *Math. Comp.*, 70(234) :827–836, 2001.

[76] A. Joux and R. Lercier. Discrete logarithms in GF(p). Email sur la mailing liste NMBRTHRY, April 2001.

[77] A. Joux and R. Lercier. Discrete logarithms in GF(p). Email sur la mailing liste NMBRTHRY, January 2001.

[78] A. Joux and R. Lercier. The function field sieve is quite special. In D.R. Kohel C. Fieker, editor, *Algorithmic number theory V*, volume 2369 of *Lecture Notes in Computer Science*, pages 431–445. Springer, Berlin, July 2002.

[79] A. Joux and R. Lercier. Improvements to the general number field sieve for discrete logarithms in prime fields. *Math. Comp.*, 72(242) :953–967, 2003.

[80] M. Joye. *Security Analysis of RSA-type Cryptosystems.* PhD thesis, Université catholique de Louvain, October 1997.

[81] B.S. Kaliski. A pseudo-random bit generator based on elliptic logarithms. In A.M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, pages 84–103, Berlin, 1986. Springer-Verlag. Lecture Notes in Computer Science Volume 263.

[82] B.S. Kaliski. One-way permutations on elliptic curves. *Journal of Cryptology*, 3 :187–199, 1991.

[83] A.A. Karatsuba and Y.P. Ofman. Multiplication of multidigit numbers on automata. *Doklady Akademii Nauk SSSR*, 145(2) :293–294, jul 1962.

[84] K.S. Kedlaya. Counting points on hyperelliptic curves using Monsky Washnitzer cohomology. *Journal of the Ramanujan Mathematical Society*, 16 :323–328, 2001.

[85] H.Y. Kim, J.Y. Park, J.H. Cheon, J.H. Park, J.H. Kim, and S.G. Hahn. Fast Elliptic Curve Point Counting Using Gaussian Normal Basis. In D.R. Kohel C. Fieker, editor, *Algorithmic Number Theory, 5th International Symposium, ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 292–307. Springer-Verlag, July 2002.

[86] D.E. Knuth. *The Art of Computer Programming : Seminumerical Algorithms*. Addison-Wesley, 2nd edition, 1981.

[87] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177) :203–209, January 1987.

[88] N. Koblitz. Hyperelliptic cryptosystems. *J. Cryptology*, 1(3) :139–150, 1989.

[89] N. Koblitz. Constructing elliptic curve cryptosystems in characteristic 2. In A.J. Menezes and S.A. Vanstone, editors, *Advances in Cryptology — CRYPTO '90*, pages 156–168, Berlin, 1990. Springer-Verlag. Lecture Notes in Computer Science Volume 537.

[90] N. Koblitz. CM-curves with good cryptographic properties. In J. Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 279–287, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science Volume 576.

[91] P.C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *Advances in Cryptology — CRYPTO '96*, pages 104–113, Berlin, 1996. Springer-Verlag. Lecture Notes in Computer Science Volume 1109.

[92] K. Koyama. Fast RSA-type schemes based on singular cubic curves $y^2 + axy = x^3$ (mod n). In L.C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, pages 329–340, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 921.

[93] K. Koyama, U.M. Maurer, T. Okamoto, and S.A. Vanstone. New public-key schemes based on elliptic curves over the ring Z_n. In J. Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 252–266, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science Volume 576.

[94] K. Kurosawa, K. Okada, and S. Tsujii. Low exponent attack against elliptic curve RSA. In *Advances in Cryptology — ASIACRYPT '94*, Lecture Notes in Computer Science, pages 376–383, Berlin, 1994. Springer-Verlag.

[95] B.A. LaMacchia and A.M. Odlyzko. Solving large sparse linear systems over finite fields. In S.A. Vanstone A.J. Menezes, editor, *Advances in Cryptology — CRYPTO '90*, pages 109–133, Berlin, 1990. Springer-Verlag. Lecture Notes in Computer Science Volume 537.

[96] S. Lang. *Algebra. 3rd revised ed.* Graduate Texts in Mathematics. 211. New York, NY : Springer., 2002.

[97] A.G.B. Lauder. Counting solutions to equations in many variables over finite fields. Preprint, 2003.

[98] A.G.B. Lauder and D. Wan. Computing Zéta functions of Artin-Schreier curves over finite fields. *LMS J. Comput. Math.*, 5 :34–55 (electronic), 2002.

[99] A.G.B. Lauder and D. Wan. Computing Zéta functions of Artin-Schreier curves over finite fields II, 2002. Preprint.

[100] A.G.B. Lauder and D. Wan. Counting rational points on varieties over finite fields of small characteristic. MSRI, Algorithmic Number Theory, 2002.

[101] A.K. Lenstra, H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261 :513–534, 1982.

[102] R. Lercier. Factoriser des entiers par la méthode des courbes elliptiques. Mémoire de DEA, July 1993.

[103] R. Lercier. Computing isogenies in $\mathbf{F}_{2^n}$. In *Algorithmic number theory (Talence, 1996)*, volume 1122 of *Lecture Notes in Computer Science*, pages 197–212. Springer, Berlin, 1996.

[104] R. Lercier. *Algorithmique des courbes elliptiques dans les corps finis.* Thèse, École polytechnique, June 1997.

[105] R. Lercier. Finding good random elliptic curves for cryptosystems defined over F_{2^n}. In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, pages 379–392, Berlin, 1997. Springer-Verlag. Lecture Notes in Computer Science Volume 1233.

[106] R. Lercier. Courbes elliptiques et cryptographie. In *Direction des Centres d'Expertise et d'Essais*, number 64 in Revue Scientifique et Technique de la Defense, pages 59–66. DGA, jun 2004.

[107] R. Lercier and D. Lubicz. Cardinality of a genus 2 hyperelliptic curve over $GF(2^{32770})$. Email sur la mailing liste NMBRTHRY, January 2003.

[108] R. Lercier and D. Lubicz. Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time. In E. Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 360–373, Berlin, May 2003. Springer.

[109] R. Lercier and D. Lubicz. A Quasi Quadratic Time Algorithm for Hyperelliptic Curve Point Counting. Journal of Ramanujan, 2004. À paraître.

[110] R. Lercier and F. Morain. Counting points on elliptic curves over F_{p^n} using Couveignes's algorithm. Rapport de Recherche LIX/RR/95/09, Laboratoire d'Informatique de l'École polytechnique (LIX), 1995.

[111] R. Lercier and F. Morain. Counting the number of points on elliptic curves over finite fields : strategies and performances. In L.C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, pages 79–94, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 921.

[112] R. Lercier and F. Morain. $\#E(GF(10^{499}+153))$. Email sur la mailing liste NMBRTHRY, January 1995.

[113] R. Lercier and F. Morain. Algorithms for computing isogenies between elliptic curves. In *Computational perspectives on number theory (Chicago, IL, 1995)*, volume 7 of *AMS/IP Stud. Adv. Math.*, pages 77–96. Amer. Math. Soc., Providence, RI, 1998.

[114] R. Lercier and F. Morain. Computing isogenies between elliptic curves over $\mathbf{F}_{p^n}$ using Couveignes's algorithm. *Math. Comp.*, 69(229) :351–370, 2000.

[115] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications.* Addison–Wesley, 1983.

[116] J. Lubin, J.-P. Serre, and J. Tate. Elliptic curves and formal groups. Available at *http :/-/ma.utexas.edu/users/voloch/lst.html*, 1964.

[117] H. Matsumura. *Commutative Ring Theory*, volume 8 of *Cambridge studies in advanced mathematics.* Cambridge University Press, 1986.

[118] K. McCurley. The discrete logarithm problem. In *Proc. Symp. in Applied Mathematics*, number 42 in Cryptology and Computational Number Theory, pages 49–74, 1990.

[119] W. Meier and O. Staffelbach. Efficient multiplication on certain nonsupersingular elliptic curves. In E.F. Brickell, editor, *Advances in Cryptology — CRYPTO '92*, pages 333–344, Berlin, 1992. Springer-Verlag. Lecture Notes in Computer Science Volume 740.

[120] A.J. Menezes, I.F. Blake, X. Gao, R.C. Mullin, S.A. Vanstone, and T. Yaghoobian. *Applications of finite fields.* The Kluwer International Series in Engineering and Computer Science. Boston : Kluwer Academic Publishers. xi, 218 p., 1993.

[121] A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5) :1639–1646, 1993.

[122] A.J. Menezes and S.A. Vanstone. Elliptic curve cryptosystems and their implementation. *Journal of Cryptology*, 6 :209–224, 1993.

[123] A.J. Menezes, S.A. Vanstone, and R.J. Zuccherato. Counting points on elliptic curves over F_{2^m}. *Mathematics of Computation*, 60(201) :407–420, January 1993.

[124] J.-F. Mestre. AGM pour le genre 1 et 2, 2001. Lettre à Gaudry et Harley. Available at *http ://www.math.jussieu.fr/˜mestre*.

[125] J.-F. Mestre. Notes of a talk given at the seminar of cryptography at Rennes, 2002. Available at *http ://www.math.univ-rennes1.fr/crypto/2001-02/mestre.ps*.

[126] B. Meyer and V. Mueller. A public key cryptosystem based on elliptic curves over Z/nZ equivalent to factoring. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, pages 49–59, Berlin, 1996. Springer-Verlag. Lecture Notes in Computer Science Volume 1070.

[127] V.S. Miller. Use of elliptic curves in cryptography. In H.C. Williams, editor, *Advances in Cryptology — CRYPTO '85*, pages 417–428, Berlin, 1986. Springer-Verlag. Lecture Notes in Computer Science Volume 218.

[128] A. Miyaji. Elliptic curves over $\mathbf{F}_p$ suitable for cryptosystems. In *Advances in Cryptology — AUSCRYPT '92*, Lecture Notes in Computer Science, pages 479–491, Berlin, 1992. Springer-Verlag.

[129] A. Miyaji. On ordinary elliptic curve cryptosystems. In *Advances in Cryptology — AUSCRYPT '92*, Lecture Notes in Computer Science, pages 460–469, Berlin, 1992. Springer-Verlag.

[130] C. Monico. The ECCp-109 challenge. Available at *http ://www.nd.edu/˜cmonico/eccp109/*.

[131] P.L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44 :519–521, 1985.

[132] F. Morain. Building cyclic elliptic curves modulo large primes. In D.W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, pages 328–336, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science Volume 547.

[133] P. Morandi. *Field and Galois Theory*, volume 167 of *Graduate texts in Mathematics*. Springer, Berlin, 1996.

[134] D. Mumford. *Tata lectures on theta I*, volume 28 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1983. With the assistance of C. Musili, M. Nori, E. Previato and M. Stillman.

[135] D. Mumford. *Tata lectures on theta II*, volume 43 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1984. Jacobian theta functions and differential equations, with the collaboration of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura.

[136] J. Neukirch. *Algebraic Number Theory*, volume 322 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1999.

[137] J. Neukirch. *Algebraic number theory*, volume 322 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1999. Translated from the 1992 German original and with a note by Norbert Schappacher, with a foreword by G. Harder.

[138] W. Nevelsteen and B. Preneel. Software Performances of Universal Hash Functions. In J. Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 24–41. Springer, 1999.

[139] NIST. *Specification for the Advanced Encryption Standard (AES)*, November 2001. Federal Information Processing Standards publication 197.

[140] J. Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, 55(192) :745–763, 1990.

[141] B. Preneel, A. Biryukov, E. Oswald, B. van Rompay, L. Granboulan, E. Dottax, S. Murphy, A. Dent, J. White, M. Dichtl, S. Pyka, Schafheutle, P. Serf, E. Biham, E. Barkan, O. Dunkelman, M. Ciet, F. Sica, L. Knudsen, and H. Raddum. Nessie security report. Technical report, NESSIE, October 2002. NES/DOC/ENS/WP5/D20/1.

[142] C. Ritzenthaler. Algorithme AGM pour les courbes de genre 3 non hyperelliptiques. Available at *http ://www.math.jussieu.fr/˜ritzenth/.*, September 2003.

[143] C. Ritzenthaler. *Problèmes arithmétiques relatifs à certaines familles de courbes sur les corps finis*. PhD thesis, Université Paris 7 - Denis Diderot, June 2003.

[144] B. Salvy, editor. *Algorithms seminar, 1994-1995*, volume 2669, chapter "Factoring Polynomials over finite fields", pages 57–60. INRIA, October 1995. Abstract of a talk given by D. Panario.

[145] T. Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.*, 15(4) :247–270, 2000.

[146] T. Satoh. On p-adic point counting algorithms for elliptic curves over finite fields. In C. Fieker and D.R. Kohel, editors, *Algorithmic Number Theory, 5th International Symposium, ANTS-V*, pages 43–66, Berlin, July 2002. Springer Verlag.

[147] T. Satoh, B. Skjernaa, and Y. Taguchi. Fast computation of canonical lifts of elliptic curves and its application to point counting. *Finite Fields and Their Applications*, 9(1) :89–101, 2003.

[148] O. Schirokauer. The function field is quite special. Preprint.

[149] O. Schirokauer. Discrete logarithms and local units. *Phil. Trans. R. Soc. Lond. A 345*, pages 409–423, 1993.

[150] A. Schönhage. Schnelle multiplikation von polynomen über körpern der charakteristik 2. *Acta Informatica*, 7(395–398), 1977.

[151] A. Schönhage and V. Strassen. Schnelle multiplication grosser zahlen. *Computing*, 7 :281–292, 1971.

[152] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of Computation*, 44 :483–494, 1985.

[153] R. Schoof. Counting points on elliptic curves over finite fields. *Journal de Theorie des nombres de Bordeaux*, 7 :219–254, 1995. Available at *http ://www.emath.fr/Maths/Jtnb/jtnb1995-1.html*.

[154] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théorie des nombres de Bordeaux*, 7 :483–494, 1998.

[155] R. Schroeppel, H. Orman, S. O'Malley, and O. Spatscheck. Fast key exchange with elliptic curve systems. In D. Coppersmith, editor, *Advances in Cryptology — CRYPTO '95*, pages 43–56, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 963.

[156] J.-P. Serre. Géométrie algébrique et géométrie analytique. *Ann. Inst. Fourier, Grenoble*, 6 :1–42, 1955–1956.

[157] D. Shanks. Class number, a theory of factorization, and genera. In *Proc. Symp. Pure Math. vol. 20*, pages 415–440. AMS, 1971.

[158] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, pages 256–266, Berlin, 1997. Springer-Verlag. Lecture Notes in Computer Science Volume 1233.

[159] J.H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986. Corrected reprint of the 1986 original.

[160] B. Skjernaa. Satoh's algorithm in characteristic 2. To appear in Mathematics of Computation, 2000.

[161] N.P. Smart. An Analysis of Goubin's Refined Power Analysis Attack. In *CHES'2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 281–290, Berlin, 2003. Springer-Verlag.

[162] C. Smith, A. et Boyd. An elliptic curve analogue of McCurley's key agreement scheme. In *Cryptography and Coding (Cirencester, 1995)*, pages 150–157, 1995.

[163] J. Stern. *Fondements mathématiques de l'informatique*. Ediscience internationale, 1990.

[164] A.L. Toom. The complexity of a scheme of functional elements realizing the multiplication of integers. *Doklady Akademii Nauk SSSR*, 4(3) :714–716, jun 1963.

[165] S.A. Vanstone. Responses to NIST's proposal. *Communications of the ACM*, 35 :50–52, 1992.

[166] F. Vercauteren. Computing Zéta functions of hyperelliptic curves over finite fields of characteristic 2. In M. Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 369–384, Berlin, 2002. Springer.

[167] F. Vercauteren. *Computing Zéta functions of curves over finite fields*. PhD thesis, Kathilieke Universiteit Leuven, November 2003.

[168] F. Vercauteren, B. Preneel, and J. Vandewalle. A memory efficient version of Satoh's algorithm. In *Advances in Cryptology — EUROCRYPT 2001 (Innsbruck)*, volume 2045 of *Lecture Notes in Comput. Sci.*, pages 1–13. Springer, Berlin, 2001.

[169] J.F. Voloch. Explicit p-descent for elliptic curves in characteristic p. *CM*, 74 :247–258, 1990.

[170] D. Weber. An implementation of the general number field sieve to compute discrete logarithms mod p. In L.C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, pages 95–105, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 921.

[171] D. Weber. Computing discrete logarithms with quadratic number rings. In K. Nyberg, editor, *Advances in Cryptology — EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 171–183. Springer–Verlag, 1998.

[172] H.C. Williams. A modification of the RSA Public Key cryptosystem. *IEEE Trans. Inform. Theory*, 6 :726–729, 1980.

Deuxième partie

Liste et résumés des publications

Liste de Publications

1 Publications postdoctorales

1.1 Journaux

- R. Lercier et D. Lubicz. A Quasi Quadratic Time Algorithm for Hyperelliptic Curve Point Counting. À paraître au Journal de Ramanujan, 2004.
- A. Joux et R. Lercier. Improvements to the general number field sieve for discrete logarithms in prime fields. *Math. Comp.*, 72(242) :953–967, 2003.
- A. Joux et R. Lercier. "Chinese & Match", an alternative to Atkin's "match and sort" method used in the SEA algorithm. *Math. Comp.*, 70(234) :827–836, 2001.
- R. Lercier et F. Morain. Computing isogenies between elliptic curves over $\mathbf{F}_{p^n}$ using Couveignes's algorithm. *Math. Comp.*, 69(229) :351–370, 2000.
- R. Lercier et F. Morain. Algorithms for computing isogenies between elliptic curves. In *Computational perspectives on number theory (Chicago, IL, 1995)*, volume 7 de *AMS/IP Stud. Adv. Math.*, pages 77–96. Amer. Math. Soc., Providence, RI, 1998.

1.2 Congrès internationaux avec comité de lecture

- R. Lercier et D. Lubicz. Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time. In G. Goos, J. Hatmanis, et J. van Leeuwen, éditeurs, *Advances in Cryptology—EUROCRYPT '2003*, volume 2656 de *Lecture Notes in Computer Science*, pages 360–373, Berlin, mai 2003. Springer.
- A. Joux et R. Lercier. The function field sieve is quite special. In D.R. Kohel C. Fieker, éditeur, *Algorithmic number theory V*, volume 2369 de *Lecture Notes in Computer Science*, pages 431–445. Springer, Berlin, juillet 2002.

1.3 Articles de vulgarisation

- R. Lercier. Courbes elliptiques et cryptographie. À paraître dans la *Revue Scientifique et Technique de la Défense*, juin 2004.
- F. Chabaud et J.-M. Couveignes, éditeurs. *Bulletin de la cellule de prospective en Codage, Cryptographie et Sécurité des Réseaux*, volume III, chapitre "Le logiciel de calcul formel MAGMA". CELAR, 1998.
- F. Chabaud et J.-M. Couveignes, éditeurs. *Bulletin de la cellule de prospective en Codage, Cryptographie et Sécurité des Réseaux*, volume II, chapitre "Courbes elliptiques et cryptographie". CELAR, 1997.

1.4 Mémoires et rapports

– F. Chabaud et R. Lercier. *ZEN, User Manual*. Disponible à `http://sourceforge.net/projects/zenfact/`, 1998.

1.5 Travaux en préparation

Livre

– H. Cohen et G. Frey, éditeurs. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, chapitre "Point Counting on Elliptic and Hyperelliptic Curves" avec D. Lubicz. et F. Vercauteren. Chapman & Hall/CRC, 2005. En préparation.

Congrès

– E. Jaulmes et R. Lercier. FRMAC, a Fast Randomized Message Authentication Code. Soumis pour publication à la conférence FSE'2005.

– S. Alt et R. Lercier. How to Make a Traceable Block Cipher Stealthy. Preprint.

2 Publications prédoctorales

2.1 Congrès internationaux avec comité de lecture

– R. Lercier. Finding good random elliptic curves for cryptosystems defined over $\mathbf{F}_{2^n}$. In *Advances in cryptology—EUROCRYPT '97 (Konstanz)*, volume 1233 de *Lecture Notes in Computer Science*, pages 379–392. Springer, Berlin, 1997.

– R. Lercier. Computing isogenies in $\mathbf{F}_{2^n}$. In *Algorithmic number theory (Talence, 1996)*, volume 1122 de *Lecture Notes in Computer Science*, pages 197–212. Springer, Berlin, 1996.

– R. Lercier et F. Morain. Counting the number of points on elliptic curves over finite fields : strategies and performances. In *Advances in cryptology—EUROCRYPT '95 (Saint-Malo, 1995)*, volume 921 de *Lecture Notes in Computer Science*, pages 79–94. Springer, Berlin, 1995.

2.2 Mémoires et rapports

– R. Lercier. *Algorithmique des courbes elliptiques dans les corps finis*. Thèse, École polytechnique, juin 1997.

– B. Salvy, éditeur. *Algorithms seminar, 1994-1995*, chapitre "Factoring Polynomials over finite fields", pages 57–60. Rapport de recherche number 2669. INRIA, octobre 1995. Résumé d'un exposé donné par D. Panario.

– R. Lercier et F. Morain. Counting points on elliptic curves over F_{p^n} using Couveignes's algorithm. Rapport de Recherche LIX/RR/95/09, Laboratoire d'Informatique de l'École polytechnique (LIX), 1995.

– R. Lercier. Factoriser des entiers par la méthode des courbes elliptiques. Mémoire de DEA, École polytechnique, juillet 1993.

Arithmétique des corps finis

– **ZEN User's manual**, F. Chabaud et R. Lercier, disponible à `http://sourceforge.net/projects/zenfact/`, 1998.

La résolution de nombreux problèmes arithmétiques nécessite la manipulation d'extensions de degré fini de $\mathbb{Z}/n\mathbb{Z}$ pour des entiers positifs n. La factorisation des nombres entiers, les tests de primalité sont des exemples de telles applications. Pour cela, les scientifiques utilisent des logiciels de calculs formels ou écrivent des programmes spécifiques (la plupart du temps en langage C).

Les logiciels de calculs symboliques manipulent avec difficulté les éléments de corps finis. Dans les pires cas, ces programmes exécutent les calculs dans les rationnels avant de finalement réduire les objets modulo n. Dans tout les cas, les applications écrites pour ces logiciels sont dix à cent fois plus lentes qu'une programmation en C. Les bibliothèques optimisées en C, par contre, ne sont souvent prévues que pour quelques anneaux finis, principalement $\mathbb{Z}/n\mathbb{Z}$.

Nous pensons que nous pouvons garder l'efficacité des bibliothèques écrites en C tout en travaillant dans n'importe quelle extension polynomiale de $\mathbb{Z}/n\mathbb{Z}$. Nous avons conçu la bibliothèque ZEN dans cet esprit.

– **FRMAC, a Fast Randomized Message Authentication Code**, E. Jaulmes et R. Lercier, en préparation.

Nous revisitons l'approche randomisée introduite dans la conception du code d'authentification de message RMAC afin de construire un MAC avec des propriétés similaires, excepté qu'il repose sur une famille de fonctions de hachage ε-universelle de type Wegman-Carter plutôt que sur une chaîne CBC. Cela conduit à un nouveau code d'authentification de message que nous appelons FRMAC dont les bornes de sécurité sont, comme pour RMAC, au delà de la limite classique du paradoxe des anniversaires. Avec des fonctions de hachage efficaces en logiciel, les performances de RMAC sont pour de grands messages similaire à celles des schémas les plus rapides connus aujourd'hui. FRMAC peut aussi être significativement plus efficace pour de petits messages. De plus, dû à des contraintes relaxées sur les vecteurs d'initialisations dans la preuve de sécurité, la mise en œuvre de FRMAC dans les applications réelles est facilitée.

Le crible algébrique

– **The function field sieve is quite special.** A. Joux et R. Lercier, ANTS V, 2002.

Dans cet article, nous décrivons des améliorations à l'algorithme Function Field Sieve (FFS) utilisé pour le calcul de logarithmes discrets dans $\mathbb{F}_{p^n}$ quand p est petit. Notre contribution principale est une nouvelle façon de construire les corps de fonctions nécessaires à l'algorithme. Avec cette nouvelle construction, la complexité heuristique est aussi bonne que la complexité de celle proposée par Adleman et Huang [AdHu99], i.e $L_{p^n}[1/3, c] = exp((c + o(1))log(p^n)^(1/3)log(log(p^n))^(2/3))$ où $c = (32/9)^(1/3)$. Avec ces deux constructions, FFS devient un équivalent du crible algébrique spécial utilisé pour factoriser des entiers de la forme $A^N \pm B$. D'un point de vue asymptotique, c'est plus rapide que des algorithmes plus anciens commme celui de Coppersmith ou la version initiale de FFS d'Adleman. D'un point de vue pratique, nous mettons en avant que cette construction a de meilleures propriétés que la construction d'Adleman et Huang. Nous démontrons l'efficacité de l'algorithme en calculant avec succès des logarithmes discrets dans un grand corps fini de caractéristique 2, i.e $\mathbb{F}_{2^{521}}$.

– **Improvements to the general number field sieve for discrete logarithms in prime fields.** A. Joux et R. Lercier, Mathematics of Computations, 2003.

Dans cet article, nous présentons de nombreuses améliorations au sujet du crible algébrique. Notre principale contribution consiste en une nouvelle façon de calculer des logarithmes individuels avec le crible algébrique sans avoir pour autant à résoudre un grand système linéaire pour chaque logarithme. Nous montrons que, avec ces améliorations, le crible algébrique est plus efficace que la méthode des entiers gaussiens à partir de modulo d'une centaine de chiffres. Nous illustrons aussi nos résultats par le calcul effectif de logarithmes discrets dans un grand corps premier.

L'algorithme SEA

– **Counting the number of points on elliptic curves over finite fields : strategies and performances.** R. Lercier et F. Morain, EUROCRYPT'95.

Les schémas cryptographiques utilisant des courbes elliptiques sur des corps finis nécessitent le calcul de la cardinalité de ces courbes. Des progrès importants ont été réalisés récemment dans ce domaine. Le but de cet article est de mettre en lumière ces améliorations et de décrire une implantation efficace de celles-ci dans le cas particulier des corps $\mathbb{F}_{2^n}$, pour $n \leqslant 500$.

– **Computing isogenies in $\mathbf{F}_{2^n}$.** R. Lercier, ANTS II, 1996.

Contrairement à ce qui se passe dans les corps finis premiers de grande caractéristique, le coût majeur lorsque l'on compte le nombre de points d'une courbe elliptique E définie sur $\mathbb{F}_{2^n}$ est le calcul d'isogénies de degré premier ℓ. La meilleure méthode connue jusqu'à présent est due à Couveignes et nécessite asymptotiquement $O(\ell^3)$ opérations dans le corps. Nous soulignons dans cet article quelques propriétés remarquables satisfaites par ces isogénies et montrons comment on peut obtenir d'elles un nouvel algorithme qui semble meilleur en pratique que celui de Couveignes avec la même complexité. Sur un problème représentatif, on gagne un facteur 5 sur le calcul total.

– **Finding good random elliptic curves for cryptosystems defined over $\mathbf{F}_{2^n}$.** R. Lercier, EUROCRYPT'97.

L'une des difficultés principales pour implanter des schémas cryptographiques basés sur des courbes elliptiques définies sur des corps finis est le nécessaire calcul de la cardinalité de ces courbes. Dans le cas de corps finis $\mathbb{F}_{2^n}$, des avancées théoriques ont amené à une accélération significative des calculs. Une fois décrit certaine de ces idées dans la première partie de cet article, nous montrons que notre nouvelle implantation fonctionne de 2 à 10 fois plus rapidement que ce qui a été fait auparavant. Dans la seconde partie, nous exhibons une légère modification de l'algorithme de Schoof pour choisir des courbes avec un nombre de points "presque premier" et ainsi des schémas cryptographiques elliptiques basés sur des courbes aléatoires au lieu de courbes spécifiques comme c'était le cas jusqu'à présent.

– **Algorithms for computing isogenies between elliptic curves.**, R. Lercier et F. Morain, Computational perspectives on number theory, 1998.

Une implantation efficace de l'algorithme de Schoof pour calculer la cardinalité de courbes elliptiques définies sur des corps finis nécessite le calcul d'isogénies entre courbes elliptiques. Nous faisons un survol des algorithmes utilisés pour réaliser cette tâche. Quand la caractéristique du corps est grande, la fonction P-Weierstrass peut être utilisée. Quand la

caractéristique du corps est petite, nous avons maintenant trois algorithmes à notre disposition, deux dus à Couveignes et un du premier auteur. Nous traitons le même exemple en utilisant ces trois algorithmes et effectuons quelques comparaisons entre eux.

– **Computing isogenies between elliptic curves over $\mathbf{F}_{p^n}$ using Couveignes's algorithm.** R. Lercier et F. Morain, Mathematics of Computations, 2000.

Le coeur des améliorations d'Elkis à l'algorithme de Schoof pour calculer la cardinalité de courbes elliptiques définies sur un corps fini est le calcul d'isogénies entre courbes. L'approche d'Elkies est bien adaptée au cas où la caractéristique du corps est grande. Couveignes a montré comment calculer des isogénies en petite caractéristique. Le but de cet article est de décrire la première implantation réussie de l'algorithme de Couveignes. En particulier, on décrit l'utilisation d'algorithmes rapides pour réaliser des opérations incrémentales sur des séries. On insiste aussi sur le cas particulier de la caractéristique 2.

– **"Chinese & match", an alternative to Atkin's "match and sort" method used in the SEA algorithm.** A. Joux et R. Lercier, Mathematics of Computations, 2001.

La méthode classique pour déterminer le nombre de points de courbes elliptiques définies sur des corps finis à partir de données partielles obtenues avec l'algorithme SEA (Schoof, Elkies, Atkin) est la méthode "Match and Sort" due à Atkin. Cette méthode est une façon de trouver via un algorithme de type "pas de bébés, pas de géants" le nombre de points parmi C candidats à l'aide de $O(\sqrt{C})$ additions sur courbes elliptiques. La méthode décrite dans cet article se débarrasse des additions sur courbes elliptiques en se servant du fait que l'on a souvent bien plus d'information au sujet du nombre de points que ce qui est réellement utilisé par la méthode d'Atkin. Cela conduit à un algorithme de complexité similaire mais l'espace nécessaire est moindre que celui nécessaire par la méthode d'Atkin. En pratique, cette méthode est bien plus efficace que celle d'Atkin puisqu'elle nous a permis de mener à bien le calcul du nombre de points d'une courbe elliptique définie sur $\mathbb{F}_{2^{1663}}$, ce qui, autant que nous le sachions, est le plus important calcul de ce type jamais réalisé. Un avantage supplémentaire est qu'il est immédiat de paralléliser ce calcul sur un réseau d'ordinateurs.

L'algorithme de Mestre

– **Counting Points on Elliptic Curves over Finite Fields of Small Characteristic in Quasi Quadratic Time.** R. Lercier et D. Lubicz, EUROCRYPT'2003.

Soit p, un petit nombre premier et $q = p^n$. Soit E une courbe elliptique définie sur $GF(q)$. Nous proposons un algorithme qui calcule sans précalculs le j-invariant du relevé canonique de E pour un coût de $O(\log n)$ fois le coût nécessaire au calcul du relevé d'une puissance du Frobenius. Soit M défini comme étant une constante telle que le produit de deux entiers de n bits peut être réalisé en $O(n^M)$ opérations élémentaires, cela conduit à un algorithme pour calculer le nombre de points d'une courbe elliptique qui atteint, au prix d'une complexité en espace de $O(n^{5/2})$, une borne théorique de complexité en temps égale à $O(n^{\max(1.19,M)+M+1/2} \log n)$. Quand le corps à une Base Normale Gaussienne de petit type, on obtient de plus un algorithme de complexité en temps égale à $O(\log(n) n^{2M})$ et en espace égale à $O(n^2)$. D'un point de vue pratique, l'algorithme correspondant est particulièrement bien adapté à une implantation. Nous soulignons cela par un calcul d'une taille de 100002 bits.

– **A Quasi Quadratic Time Algorithm for Hyperelliptic Curve Point Counting.** R. Lercier et D. Lubicz, à paraître au Journal de Ramanujan, 2004.

Nous décrivons un algorithme pour calculer la cardinalité de jacobiennes de courbes hyperelliptiques ordinaires de petit genre définies sur des corps finis $\mathbb{F}_{2^n}$ avec complexité $O(n^{2+o(1)})$. Cet algorithme dérive d'idées dues à Mestre. Plus précisément, nous donnons les éléments mathématiques derrière l'algorithme de Mestre et développons à partir de celui-ci une variante avec complexité quasi-quadratique. Entre autres choses, nous présentons un algorithme pour trouver les racines d'un système d'équations d'Artin-Schreier généralisées et donnons les résultats que nous obtenons avec une implantation efficace. En particulier, nous avons pu calculer la cardinalité de courbes de genre un, deux ou trois définies sur des corps finis de taille gigantesque.